

VŠB – Technická univerzita Ostrava
Fakulta elektrotechniky a informatiky

BAKALÁŘSKÁ PRÁCE

VŠB – Technická univerzita Ostrava
Fakulta elektrotechniky a informatiky
Katedra měřicí a řídicí techniky

Vestavěný systém pro monitorování a analýzu obrazu
Embedded System for Figure Monitoring and Analysis

Zadání

Student: Aleš Holaň
Studijní program: B2645 Elektrotechnika, sdělovací a výpočetní technika
Studijní obor: 2612R041 Řídící a informační systémy
Téma: Vestavěný systém pro monitorování a analýzu obrazu
Akademický rok: 2009/2010

Zásady pro vypracování:

1. Rozbor problematiky měření a zpracování obrazu z kamery.
2. Návrh a realizace vestavného systému pro snímání a analýzu obrazu.
3. Řešení systému pro analýzu obrazu s uživatelským prostředím.
4. Srovnání naměřených, analyzovaných výsledků s teoretickými předpoklady.
5. Zhodnocení dosažených výsledků.

Seznam doporučené odborné literatury:

1. VÍT, V.- KUBA, P. Televizní technika: studiové zpracování televizního signálu. 1.vyd. Praha: BEN technická literatura, 2000. 208 s. ISBN 80-86056-88-0.
2. VÍT, V. Televizní technika: přenosové barevné soustavy . 1.vyd.Praha: BEN technická literatura, 1997. 720 s. ISBN 80-86056-04-X.
3. VÍT, V.- GREGORA, P. Televizní technika: Zařízení pro přenos a vysílání televizního signálu. 1.vyd.Praha : BEN technická literatura, 2000. 176 s. ISBN 80-86056-89-9.
4. BURKHARD, K.USB: měření, řízení a regulace pomocí sběrnice USB. 1. vyd.Praha : BEN technická literatura, 2002. 256 s., CD. PC & elektronika. ISBN 80-7300-073-3.
5. BEZDĚK , M. Elektronika III. 1.vyd. [s.l.] : Kopp, 2004. 236 s. ISBN 80-7232-241-9.

Vedoucí: Ing. Zdeněk Macháček, Ph.D.

Datum zadání: 20.11.2009

Datum odevzdání: 7.5.2010

PODĚKOVÁNÍ

Děkuji vedoucímu mé bakalářské práce Ing. Zdeňkovi Macháčkovi, Ph.D. za pomoc a podporu při tvorbě bakalářské práce.

PROHLÁŠENÍ

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

Datum

Podpis

ABSTRAKT

Tato bakalářská práce je zaměřena na seznámení se s principy snímání obrazu pomocí analogových černobílých videokamer. Tyto znalosti jsou poté využity k sestavení prvků a algoritmů pro snímání obrazu z analogové kamery pomocí vestavného systému pro monitorování obrazu a jejich přeposílání do počítače, kde se provádí skládání, vyhodnocování a další analýza obrazu.

Analýza obrazu se v dnešní době využívá v mnoha odvětvích průmyslu od zabezpečovacích aplikacích pro zajištění bezpečnosti na pracovišti až po kontrolu jakosti výrobků. Také ji můžeme využívat pro laboratorní úlohy či pro řízení mobilních robotů, rozpoznávání tváří, snímání otisků prstů nebo vyhledávání textu či tvarů v obraze. V této práci se analýza zaměřuje na vyhledávání spojitostí v obraze a jejich porovnávání s předem známými znaky jako jsou číslice nebo písmena.

ABSTRACT

This Bachelor thesis is focused on familiarizing oneself with the methods of image acquisition using analog black-and-white cameras. This knowledge is then applied to building elements and algorithms that utilize captured images from analog cameras via an embedded image monitoring system. These images are then forwarded to the computer, assembled, evaluated and further analyzed.

The image analysis is already being used in many branches of industry from security applications to ensure safety in the workplace, to the quality control of products. It is also possible to use it for laboratory tasks, mobile robot control, facial recognition, fingerprints scanning or to recognize text or specific shapes in an image. This thesis' analysis focuses on searching for continuities in an image and comparing them with previously known characters such as digits or letters.

KLÍČOVÁ SLOVA

obraz, analýza obrazu, kamery, snímání obrazu, korekce obrazu, jasové transformace, negace, gama korekce, prahování, histogram

KEYWORDS

image, image analysis, cameras, image acquisition, image correction, luminance transformations, negation, gamma correction, thresholding, histogram

Seznam použitých symbolů a zkratek

CMOS	Complementary Metal Oxid Semiconductor
COM	rozhraní mezi počítačem a dalším zařízením
UART	Synchroní a asynchroní seriové rozhraní
USB	Universal Serial Bus, universální seriová sběrnice
A/D	Analogově-digitální převodník
FIFO	First In First Out
EEPROM	Electrically Erasable Programmable Read-Only Memory
RS232, RS458	komunikační rozhraní

Obsah

1	ÚVOD	1
2	Základní principy snímání obrazu	2
2.1	Snímací elektronka typu Vidikon.....	2
2.2	Snímací elektronka typu Plumbikon	3
3	Navrhovaná struktura zařízení	5
3.1	Kamera T846 B/W CMOS	5
3.1.1	Parametry kamery	5
3.1.2	Měření videosignálu na analogové kameře T846	6
3.1.3	Analýza měřeného videosignálu	8
3.2	Univerzální vývojová deska EvbDSP56800	9
3.2.1	A/D převodník	10
3.2.2	Sériová komunikace SCI	10
3.2.3	Časovač	10
3.3	USB modul.....	10
4	Monitorování videosekvence	13
4.1	Vzorkování signálu pomocí A/D převodníku	13
4.2	Zajištění stability a posunu v obraze	14
4.3	SCI komunikace v EvbDSP56800	16
4.4	Příjem dat v počítači.....	17
5	Zpracování obrazové videosekvence	18
5.1	Transformace jasu obrazu	18
5.1.1	Transformace obrazu na negativ	19
5.1.2	Gama korekce obrazu	20
5.1.3	Roztažení kontrastu obrazu.....	21
5.1.4	Prahování obrazu	23
5.2	Histogram obrazu	24
6	Analýza videosekvence.....	26
6.1	Metoda automatické hledání prahu	26
6.2	Detekce spojitostí v obraze	29
6.3	Analýza znaků v obrazu	30
6.3.1	Segmentace objektu	30

6.4	Uživatelské rozhraní.....	34
6.5	Měření a vyhodnocení výsledku analýzy obrazu	36
7	Závěr	39
8	Seznam použité literatury	41
9	Seznam příloh	42

1 ÚVOD

Tato bakalářská práce je zaměřena na vytvoření a sestavení zařízení schopného monitorovat obraz snímáný pomocí analogové kamery. Zařízení je založeno na univerzální vývojové desce EvbDSP56800 a omezených parametrech této desky. Bakalářská práce se zaměřuje na způsoby vyhledávání znaků v obraze, především číslic nebo písmen. Jsou zde také ukázány minimální nutné parametry potřebné pro kvalitní snímání obrazu z analogových kamer a jeho následnou digitalizaci.

Současný rozvoj spotřební elektroniky v oblasti fotoaparátu a videokamer se zaměřuje na širokou veřejnost. Díky tomu trh umožňuje vytváření levných a zároveň spolehlivých technických aplikací pro různé obory technické praxe. V technické praxi se například jedná o kontrolu jakosti produktů, sledování a řízení množství průmyslových a laboratorních procesů, sledování a kontrolu provozu na komunikacích nebo identifikace osob nebo objektů. Ale také pro zajištění bezpečnosti na pracovišti a hlídání objektů.

Dva principy snímání obrazu pomocí analogových videokamer jsou popsány v kapitole 2. A to jeden z prvních snímacích prvků Vidikon a jeden s pokročilejších snímacích prvků Plumbikon.

3. kapitola je zaměřena na popis videosignálu z analogové kamery a používaného zařízení včetně základních použitých prvků, které jsou využívány na univerzální vývojové desce EvbDSP56800 jako jsou časovače, A/D převodník, SCI komunikace. Dále je zde popsán USB modul navrhnutý a zkonstruovaný pro tuto práci.

Kapitola 4 Monitorování videosekvence. se zaměřuje na popis propojení jednotlivých prvků, jejich vzájemnou komunikaci a je zde popsáno nastavení a program vestavného systému pro monitorování obrazu.

Základní zpracování obrazu videosekvence je popsáno v kapitole 5. Zde jsou popsány jednoduché transformace jasu sloužící pro zlepšení obrazu, ale také histogram využívaný při některých transformacích jasu nebo také používán při hledání optimálního prahu.

V kapitole 6 Analýza videosekvence je popsána metoda vyhledávání optimálního prahu sloužícího pro vytvoření prahovaného obrazu, který je poté analyzován. Dále je zde popsán princip detekce spojitostí v obraze a analýza znaků v obraze. Také popisuje uživatelské rozhraní programu napsaného přímo pro tuto bakalářskou práci.

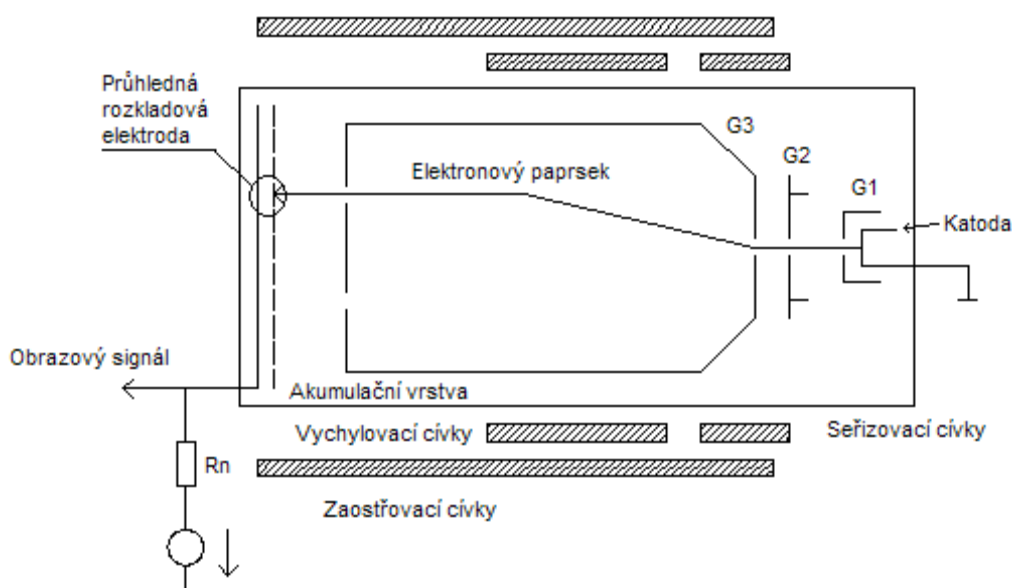
Zhodnocení použitých postupů a jejich vyhodnocení je provedeno v 7. kapitole.

2 Základní principy snímání obrazu

Snímání obrazu je založeno na odrazu světla od překážky a jeho pohlcování fotocitlivým prvkem. Jsou zde uvedeny dva základní principy u analogových kamer, a to jeden z prvního principu snímacích elektronek Vidikon a dále modernějšího principu jako je elektronky Plumbikon od firmy Philips.

2.1 Snímací elektronka typu Vidikon

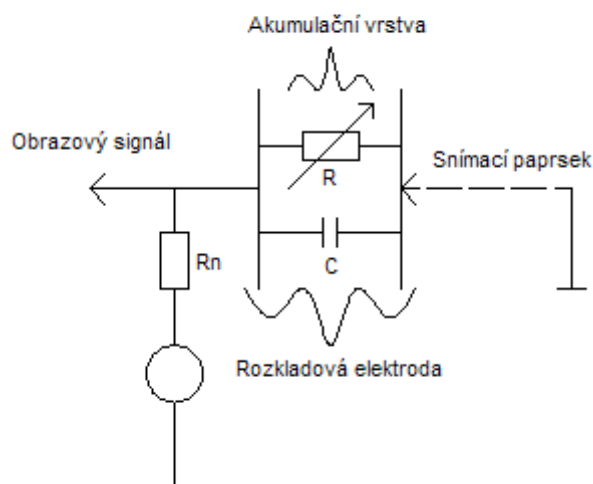
Vidikon spadá mezi jednoduché snímací elektrony, které obsahují pouze vychylování elektronového paprsku a rozkladovou elektronku. Elektronka je složena z průhledné rozkladové elektrody napařené na vnitřní straně čelní průhledné desky trubice, na niž je směrem do vnitřního prostoru nanесena fotoelektricky citlivá akumulací vrstva.



Obr. 2.1 Základní princip snímání Vidikonu [1]

Obr. 2.1 zobrazuje základní princip snímací elektrony typu Vidikon. Snímací paprsek (elektronový paprsek) je generován elektronovou tryskou. Pomocí elektrody G1 je řízena proudová hustota snímacího paprsku a poté je urychlován pomocí elektrody G2. Následně je pomocí elektrody G3 snímací paprsek zaostřován. Seřizovací cívky slouží pro korigování směru paprsku po opuštění elektronové trysky.

Vychylovací cívky slouží pro nasměrování snímacího paprsku na daný bod. Aby bylo zajištěno kolmé dopadu paprsku na rozkladovou elektrodu je upraveno podélným magnetickým polem zaostřovacích cívek. Snímací paprsek je ještě před dopadem na rozkladovou elektrodu zpomalován elektrostatickým polem mezi signální elektrodou a stínítkem uzavírajícím válcovou elektrodu G3.



Obr. 2.2 Princip jednoho prvku vidikonu [1]

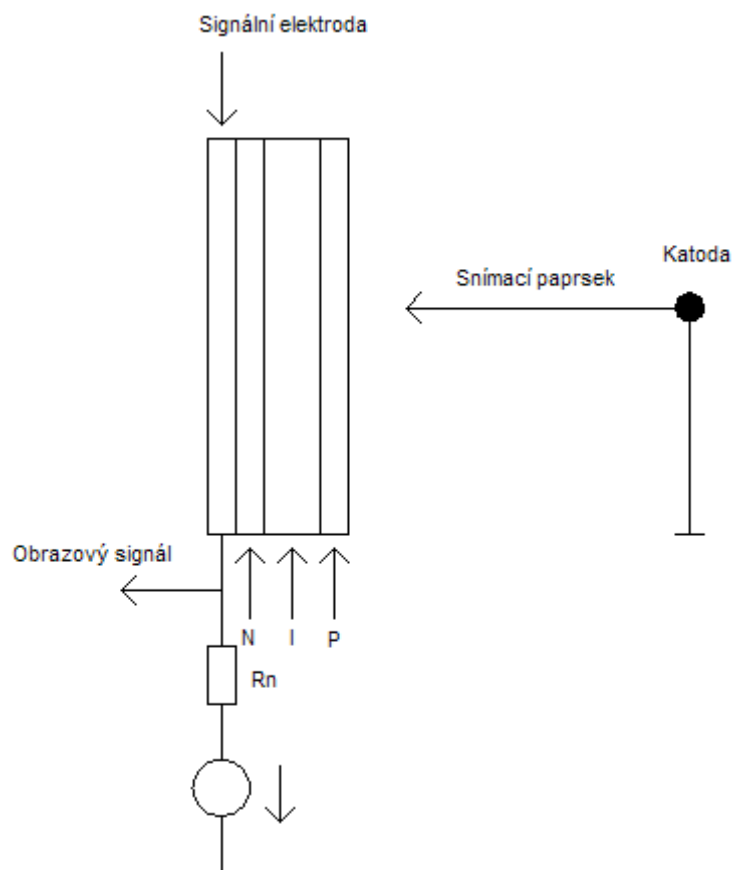
Pro lepší pochopení snímání obrazu elektronkami typu vidikon si lze představit jeden obrazový bod (pixel) jako paralelní zapojení proměnlivého odporu a kapacity (Obr. 2.2). Snímací paprsek postupně prochází celou plochou akumulací vrstvy. V důsledku dopadu snímacího paprsku se kapacita nabíjí na potenciál snímací elektrody a po celou dobu snímací periody se vybíjí přes proměnlivý odpor, jehož okamžitá hodnota je závislá na velikosti osvitu akumulací vrstvy. Při následném snímání dochází k dobití kapacity na potenciál rozkladové elektrody a přebytečný náboj vytváří na odporu R_n samotný obrazový signál.

Mezi nevýhody vidikonu patří, například setrvačnost a nerovnoměrná úroveň černé. Nerovnoměrnost černé je způsobena neideálním rozložením tloušťky akumulací vrstvy, což má za následek různé parametry odporů v jednotlivých buňkách. Setrvačnost je asi největším nedostatkem vidikonu, v důsledku toho dochází k neostrosti pohybujících se předmětů.

[1]

2.2 Snímací elektronka typu Plumbikon

Plumbikon patří mezi snímací elektronky stejně jako vidikon. Na rozdíl od vidikonu nemá některé nevýhody jako je setrvačnost. Nevýhoda byla odstraněna použitím jiného materiálu rozkladové elektrody. U plumbikonu je jako rozkladová elektroda používán závěrně polarizovaný polovodičový přechod PIN, přičemž P a N jsou polovodiče s nevlastní vodivostí a I je polovodičový materiál s vlastní vodivostí. Díky závěrně polarizovanému přechodu jsou potlačeny proudy za tmy, a tím nedochází k rušivým signálům při úrovni černé barvy.



Obr. 2.3 Základní princip plumbikonu[1]

Povrch P akumulární elektrody je pomocí elektronového paprsku stabilizován na zemní potenciál katody. Díky závěrnému přepětí přechodu PIN se chová jako fotodioda, v níž se při dopadu světla tvoří volné páry elektron-díra. Díry pronikají do vrstvy P, zatím co signální elektroda k sobě přitahuje volné elektrony. To vytváří, na rozkladové elektrodě, potencionální reliéf snímané scény. Pomocí snímacího paprsku se poté připojují jednotlivé snímané body na zem, čímž se obvod uzavírá pro proud neutralizující vzniklý náboj a ten vytváří na odporu R_n obrazový signál (Obr. 2.3).

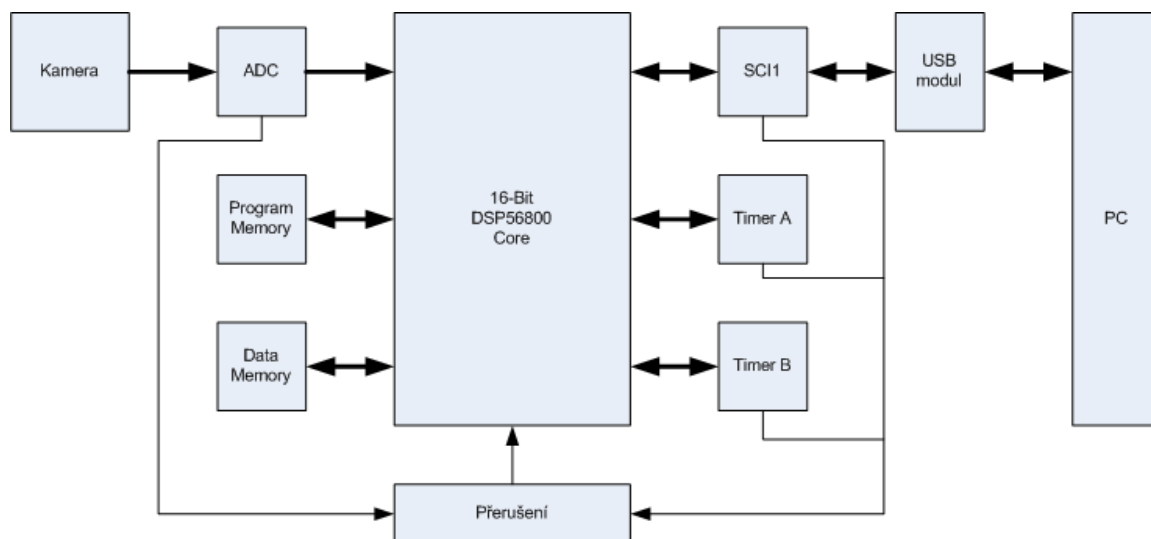
[1]

3 Navrhovaná struktura zařízení

Pro účely této bakalářské práce bylo nutno navrhnout zařízení schopné snímat obraz. Jako prvním blok na Obr. 3.1 je možno vidět kameru. Nejdříve bylo nutno zjistit základní informace o této kameře. Z tohoto důvodu na ni muselo být provedeno měření. To je popsáno v podkapitole 3.1 Kamera T846 B/W CMOS.

Signál z kamery je nutno nejdříve digitalizovat. Jako řídicí modul byla použita laboratorní vývojová deska označená EvbDSP56800 VŠB-TOU, která je osazena signálovým procesorem firmy Freescale DSP56F805. Tato část zařízení je blíže popsána v podkapitole 3.2 Univerzální vývojová deska EvbDSP56800.

Celé zařízení je připojeno na komunikační rozhraní USB pomocí USB modulu, který umožňuje odesílat získaná data přímo do počítače. Modul je navrhnut a realizován přímo pro účely této aplikace a je založen na čipu FT232RL od firmy FTDI. Ten je blíže rozepsán v podkapitole 3.3 USB modul.



Obr. 3.1 Blokové schéma zařízení

3.1 Kamera T846 B/W CMOS

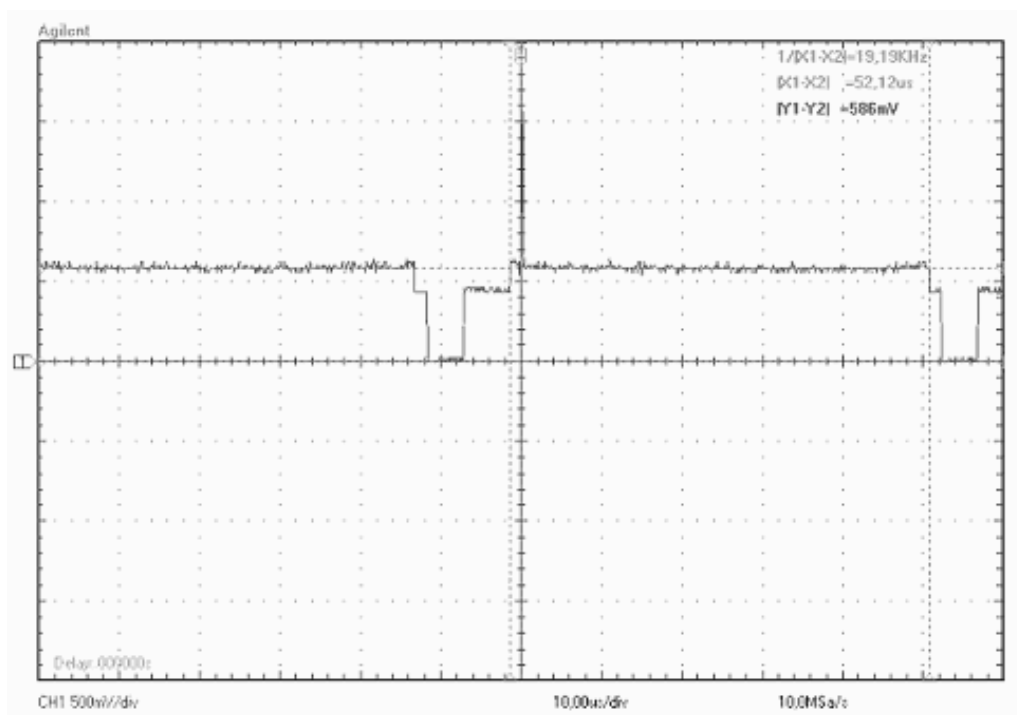
3.1.1 Parametry kamery

Senzor CMOS 1/3", obrazová plocha 5,78 x 4,19 mm, 628 x 582 pixel, 320 řádků, video výstup: 1V p-p/75 ohm, auto řízení zisku 18 dB, osvětlení diodami infra - 6x, automatická závěrka 1/60 - 1/15 000; s audio - elektret. mikrofón, rozměr 20 x 20 x 14 mm, napájení 6-9 V/30mA

3.1.2 Měření videosignálu na analogové kameře T846

Účel měření videosignálu bylo zjistit, pomocí osciloskopu DSO3062A, základní informace o černobíle kameře T846. Pro vhodné zpracování informací lze osciloskop připojit pomocí rozhraní USB k počítači. Díky povaze signálu je nutno celé měření provádět v režimu Single.

První měření bylo provedeno s krytkou, čímž je možno dosáhnout na snímacím prvku takřka absolutní tmu, a tím také skoro ideální černou barvu. Na Obr. 3.2 je vidět jeden řádek obrazového signálu ze snímače kamery. Samotné body řádku jsou omezeny vertikálními kurzory, které naměřily přibližně délku trvání vykreslované části $52\mu\text{s}$. Také je možno změřit hodnotu zatemňovací úrovně (hodnota je $4,4\mu\text{s}$) celkovou délku řádkového impulsu včetně zatemňovací úrovně, ten má hodnotu $64\mu\text{s}$. Velikost signálu mezi vertikálními kurzory je rovna úrovni černé barvy, jenž je omezena horizontálními kurzory, přičemž hodnota černé barvy je tedy 586mV . Ze stejného zachyceného signálu lze také zkontrolovat hodnotu stejnosměrné složky, která je nižší a její hodnota je pouze 450mV . Z teorie televizního signálu a poznatků získaných při měření této kamery vyplývá několik skutečností, které mohou pomoci při vytváření samotné aplikace.



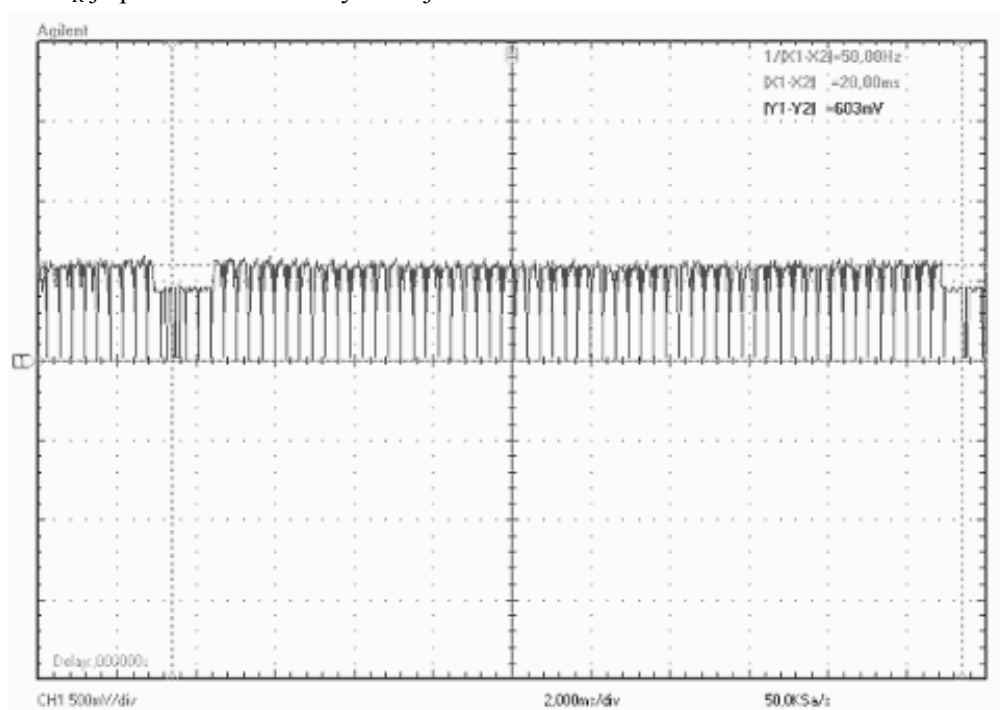
Obr. 3.2 Jeden řádek obrazového signálu při hodnotě černé barvy

Pak tedy lze poznat, že i při úplné tmě, je úroveň černé barvy ze snímače vyšší než stejnosměrná složka signálu. S toho vyplývá, že při další analýze signálu se bude muset tato hodnota brát v úvahu.

Za předpokladu, že v užitečné části signálu se skrývá 520 bodů (pixel) na jednom řádku, se dá vypočíst minimální vzorkovací frekvence na 10MHz to vyplývá z naměřených hodnot podle vztahu:

$$f_s = \frac{P_R}{T_R} = \frac{520}{52 \cdot 10^{-6}} = 10\text{MHz} \quad 3.1$$

- Kde
- f_s je potřebná frekvence vzorkování
 - T_R je časové trvání užitečné části řádku
 - P_R je počet bodů snímaných na jednom řádku



Obr. 3.3 Zobrazený půl snímek televizního signálu

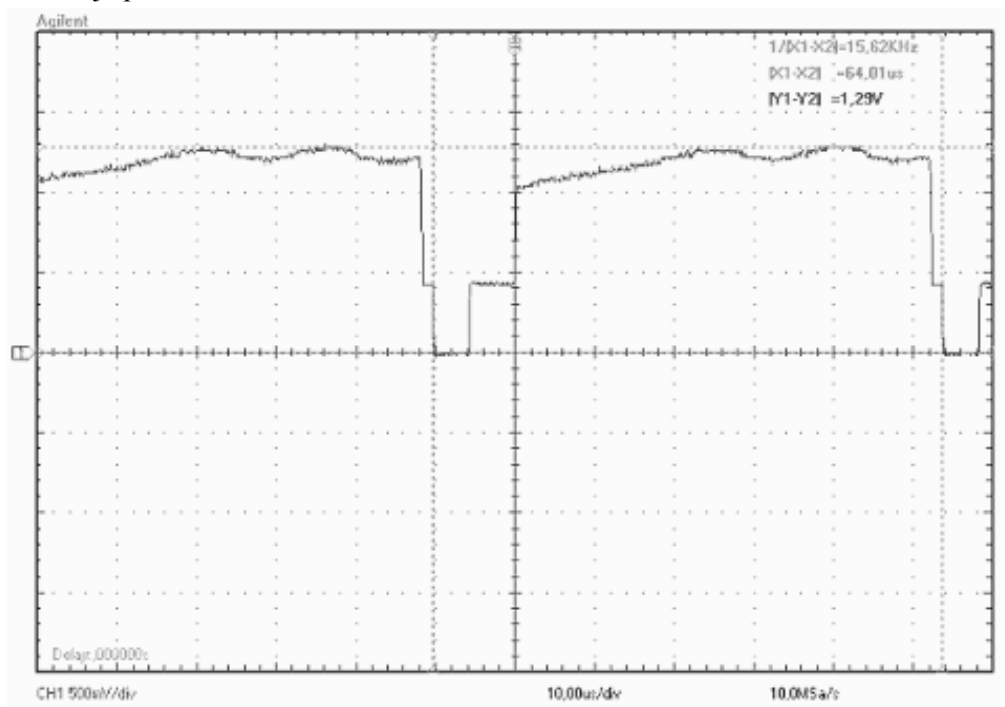
Další změřenou informací byl změřený půl snímek televizního signálu, který je zobrazen na Obr. 3.3. Délka jednoho půl snímku je tedy 20ms, což odpovídá frekvenci 50Hz. Z teorie televizního signálu je možno zjistit, že pro plynulé vnímání obrazu lidským okem je zapotřebí minimálně 50 snímků za sekundu (neboli frekvence 50Hz). Kvůli vysokým frekvencím, které by se musely používat pro modulaci takového signálu, se začaly přenášet pouze takzvané půl snímky s frekvencí 50 půl snímků za sekundu. Půl snímek znamená, že se nejdříve odešlou v prvním snímku liché řádky a následně v dalším snímku se teprve odešlou sudé řádky, ale díky nedokonalosti lidského oka dochází ke stejnému efektu jako kdyby výstupní obraz obsahoval všechny řádky s frekvencí 50Hz.

Z naměřených hodnot vyplývá statická délka jednoho řádku obrazového signálu na 64μs a délku jednoho půl snímku na 20ms. Poté lze vypočítat maximální počet řádků v jednom půl snímku.

$$R_M = \frac{T_S}{T_R} = \frac{20 \cdot 10^{-3}}{64 \cdot 10^{-6}} = 312,5 \quad 3.2$$

Kde

- R_M je maximální počet řádků
- T_S je perioda půl snímku
- T_R je perioda řádku



Obr. 3.4 Jeden řádek obrazu při relativní bílé barvě

Problém nastává při snaze měřit výstupní signál kamery při absolutní bílé barvě. Bílá barva je totiž závislá na barvě a velikosti osvětlení, čistoty podkladu, odrazových vlastnostech a dalších faktorech. Toto měření není přesné, ale může navést na správnou cestu při další analýze dat. Toto měření proběhlo při záměrném mírném přesycení snímáče výrazným zdrojem světla. Díky přesycení snímáče dochází k vygenerování maximálního výstupního signálu z kamery. Z Obr. 3.4 je možno vyčíst hodnotu relativní bílé barvy ta je tedy přibližně 1,3V.

3.1.3 Analýza měřeného videosignálu

Analýzou měřených dat lze zjistit zásadní informace pro návrh částí aplikace a také pro následnou analýzu obrazu z kamery.

Pro zajištění maximálního rozlišení je nutno dodržet vzorkovací frekvenci f_s (podle rovnice 3.1), alespoň 10MHz. Toho je možno dosáhnout, například paralelním A/D převodníkem. Z Shannon-Kotelnikova teorému vyplývá [7], že pro zajištění zpětné rekonstrukce signálu musí být vzorkovací frekvence minimálně dvojnásobná než maximální harmonická složka signálu a z tohoto důvodu je nutno před A/D předřadit dolnoproustný filtr

s nejvyšším mezním kmitočtem 5MHz, čímž je dosaženo odříznutí vyšších harmonických složek signálu.

Na Obr. 3.3 je možno vidět že půl snímek obsahuje, krom řádkových impulsů, také začáteční a koncové oddělení jednotlivých půl snímků. Za předpokladu, že jeden půl snímek obsahuje pouze 300 obrazových řádků, pak je možno určit potřebnou minimální paměť pro uložení jednoho půl snímku podle rovnice:

$$M = R \cdot P_R = 520 \cdot 300 = 156000 B = \frac{188400}{2^{10}} \cong 153kB \quad 3.3$$

Kde

- M je velikost paměti
- R je počet řádků
- P_R je počet bodů snímaných na jednom řádku

Ovšem v případě uchování celého půl snímku včetně přebytečných částí, tedy zatemňovacích úrovní, začátku a konce půl snímku je nutno pro výpočet minimální potřebné paměti využít jiného vzorce vyplývajícího se znalosti vzorkovací frekvence a statické délky půl snímku:

$$M = \frac{T_s}{\frac{1}{f_s}} = T_s \cdot f_s = 20 \cdot 10^{-3} \cdot 10 \cdot 10^6 = 200000 B = \frac{200000}{2^{10}} \cong 196kB \quad 3.4$$

Kde

- M je velikost paměti
- T_s je perioda půl snímku
- f_s je frekvence vzorkování

Pro následnou analýzu lze využít faktu, že stejnosměrná složka je nižší než skutečná hodnota černé barvy. To může pomoci v zredukování potřebné paměti pro ukládání hodnot vyšších než je stejnosměrná složka a nižších než je hodnota černé barvy.

3.2 Univerzální vývojová deska EvbDSP56800

Univerzální vývojová deska EvbDSP56800 je založena na signálovém mikroprocesoru DSP56F805 od firmy Freescale. Tato deska je sestavena a vytvořena pro laboratorní účely přímo firmou Freescale. Obsahuje mnoho prvků, jako je například vstupně-výstupní porty, PWM výstupy, A/D vstupy a mnoho dalších prvků.

Pro programování signálového mikropočítače DSP56F805 je využito integrované vývojové prostředí (IDE) CodeWarrior založeného na programovacím jazyce C. Nastavování je prováděno pomocí nástroje pro návrh aplikací Processor Expert od firmy Freescale. Je zde využito takzvaných Beans, které slouží právě pro nastavování registrů signálových mikropočítačů bez jejich předchozí důkladných znalostí.

3.2.1 A/D převodník

Ve vyvinuté aplikaci je použit vnitřní A/D převodník, na který je přiváděn signál přímo z kamery.

DSP56F805 obsahuje 8 analogových vstupů, které jsou přivedeny na dva oddělené multiplexery, ty slouží pro připojení jednoho z analogových vstupů dále na A/D převodník (dva mux obvody jsou využity kvůli zmenšení případné chyby). Následně je signál z daného analogového vstupu přiveden na obvody sample/hold a ty slouží pro vzorkování signálu, jež je následně ještě kvantován a kódován. Samotný A/D převodník je 12 bit sériový převodník s maximální vzorkovací periodou 1,7μs.

3.2.2 Sériová komunikace SCI

Pro odesílání získaných dat je využito sériové rozhraní SCI. Jedná se o asynchronní komunikaci. Díky čemuž je zde možná plně duplexní komunikace. Plně duplexní komunikace znamená, že obě zařízení připojena na tuto sběrnici a mohou zároveň přijímat i odesílat data.

U SCI komunikace je využito pouze dvou vodičů, a to RXD a TXD, a nedochází k vysílání žádných synchronizačních impulsů, proto musí být na obou zařízeních nastavena stejná rychlost, které se určuje v tzv. baud, což je jednotka modulační rychlosti (anglicky baud rate). Tato rychlost odpovídá zhruba poměru 1 baud = 1 bit/s (u sběrnice RS232). DSP56F805 obsahuje 2 SCI sběrnice. Pro tuto aplikaci je využito SCI1 a rychlost byla zvolena na 825000 baud, tedy 825000 bit/s. Tato rychlost byla zvolena kvůli náročnosti aplikace na celkovou rychlost.

3.2.3 Časovač

Časovač (anglicky timer) je zařízení schopné odměřovat čas v přesných intervalech. Slouží pro synchronizaci některých procesů, jako je například v tomto případě pro zajištění pravidelného načítání signálu z A/D převodníku, a tím také pravidelného načítání videosignálu z analogové kamery. Činnost časovače je založena na čítání impulsů oscilátoru používaného u daného mikropočítače. V případě, že dojde k načtení určitého počtu impulsů, je vygenerováno přerušení, a to vykoná příslušnou funkci. V případě této aplikace je to zapnutí A/D převodníku. Od oscilátoru se také odvíjí maximální rychlost a přesnost časovače. To může mít za následek kolísání nebo nepřesnost vykonávání přerušení v případech delších časových intervalů. DSP56F805 obsahuje čtyři základní čítače podporující více funkcí. Je využito jednoho čítače s periodou 40ms, čímž je zajištěno pravidelné snímání stejných bodů ve videosignálu analogové kamery. Druhý časovač je nastaven na 800ms a slouží pro spínání sekvence načtení snímku.

3.3 USB modul

Tento modul je založen na integrovaném obvodu FT232RL firmy FTDI, čímž umožní velmi snadné připojení uživatelské aplikace k počítači pomocí sběrnice USB a díky ovladačům napsaným přímo k tomuto čipu, není zapotřebí si psát vlastní ovladače k aplikaci. Modul je

Na Obr. 3.5 je možno vidět schéma zapojení modulu převodníku USB – UART. Vstupem je přímo USB konektor typu B. K napájení čipu je použito napájení z aplikace.

Výrobce poskytuje dva typy ovladačů D2XX a VCP. D2XX je ovladač, který umožňuje přístup přes USB Stack Windows a DLL knihovny, díky čemuž je možno přistupovat k aplikaci přes programovací jazyky Visual C#, Visual C++ a další. Ovladač VCP (Virtual Com Port) vytváří virtuální com port, ke kterému lze přímo přistupovat například z Matlabu a komunikovat přes něj z USB modulem a dále z uživatelské aplikace. Identifikace VID a PID jsou informace potřebné k vyhledání příslušného ovladače zařízení a ovladač následně požádá o uvolnění potřebných prostředků počítače. Právě ovladač je zodpovědný například za uvolnění většího proudu na sběrnice USB (to znamená v případě potřeby umožnit zařízení odebírat potřebný proud do maximální hodnoty 500mA).

Integrovaný obvod FT232RL je prakticky kompletní převodník USB – UART. Z několika programovatelnými vstupně-výstupními porty a z možností různých způsobů přenosu dat. Jak je možno vyčíst z technické dokumentace čip obsahuje dvě paměti FIFO, vstupní (FIFO TX Buffer), která má velikost 256 bytu a výstupní (FIFO RX Buffer), ta je 128 bytová. Obsahuje také interní EEPROM paměť pro uložení Vendor ID (VID – informace o výrobci), Produkt ID (PID – informace o produktu)[3], sériové číslo zařízení a další informace. EEPROM paměť lze programově nastavit podle požadavků aplikace. Jeho výhodou je také vnitřní oscilátor, díky kterému nepotřebuje vnější zdroj oscilátoru. Dále obsahuje regulátor napětí na 3,3V, který je schopen do uživatelské aplikace dodávat 50mA. Také je možno vidět na blokovém schématu USB protokol engine, SIE (serial interface engine) a UART FIFO Controller. Ty slouží pro samotný převod se sběrnice RS232 (popřípadě RS458) na sběrnici USB.

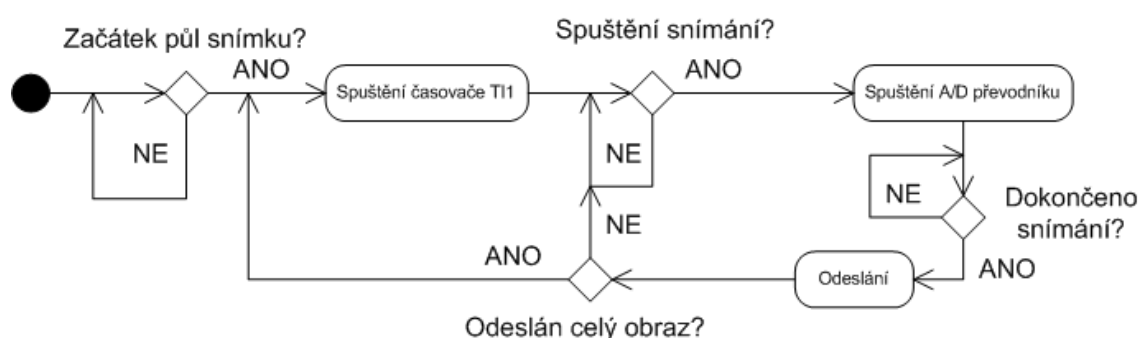
Modul je k počítači připojen přímo přes USB sběrnici. K multifunkční desce je modul připojen přímo na port D. Port D je využíván u mikroprocesoru jako vstupně-výstupní port. Všechny pinů portu D jsou vyvedeny na desku plošného spoje. Kromě vstupně-výstupního režimu jde port D využít také pro sériovou komunikaci. Na pin GPIOD7 a GPIOD6 je vyveden výstup a vstup sériového portu 1. Na výstup RXD1 mikroprocesoru je připojen vstup TXD USB modulu a na vstup TXD1 mikroprocesoru je připojen výstup RXD USB modulu, což je asynchronní sériový přenos. Asynchronní přenos je duplexní přenos, ten se vyznačuje možností vysílat a přijímat data současně a nejsou vyžadovány synchronizační impulsy. Na rozdíl od synchronního přenosu, kde jsou propojeny vstupy TXD1 a TDX mezi sebou, a RXD1 by byl propojen s RXD, potom se přes propojený výstup posílají synchronizační impulsy, které říkají zařízení, kdy má přečíst data ze vstupů nebo kdy má data vyslat. Tento režim je pouze částečně duplexní. Dále jsou na port D připojeny některé ze signalizačních vstupů a výstupy USB modulu. Výstup RTS je výzva k vysílání ten je připojen na vstup GPIOD5. Vstup CTS připojen na výstup GPIOD4 je pohotovost k vysílání.

4 Monitorování videosekvence

Pro zachycování obrazu je využito analogové videokamery. Spojitý signál je nutno nejdříve digitalizovat. Pro digitalizaci spojitého signálu je využito vnitřního omezeného A/D převodníku univerzální desky EvbDSP56800 s maximální vzorkovací periodou 1,7 μ s. Tento digitalizovaný signál ovšem obsahuje i přebytečné části, jako jsou zatemňovací úrovně oddělující jednotlivé řádky signálu či určování začátků jednotlivých půl snímků.

Univerzální deska EvbDSP56800 vykonává v této aplikaci vzorkování spojitého signálu a jeho odesílání dále do počítače. Také je na ni implementována jednoduchá funkce pro výběr prahu pomocí výběru středu mezi maximální a minimální hodnotou v signálu. Tento obraz je také odeslán do počítače pro vykreslení.

Počítač má úlohu v této aplikaci složení obrazu a jeho komplexnější analýzu.



Obr. 4.1 Diagram hlavního programu zařízení

4.1 Vzorkování signálu pomocí A/D převodníku

I když podle rovnice 3.1 vyplývá minimální frekvence pro vzorkování 10MHz, toto se dá obejít pomocí postupného sběru dat. Tedy je sejmuto při prvním průchodu například každý desátý bod obrazu. Následně, když nastává další opakování obrazu (každých 40ms), musí dojít k časovému posunu odpovídajícímu délce trvání jednoho obrazového bodu (tedy 100ns). Tento princip lze využít pouze za předpokladu stabilního zaměření na daný objekt, jestliže toto pravidlo nebude dodrženo, může docházet k deformacím obrazu.

Pro navržení programu v Processor Expert bylo nutno připojení dvou tzv. Beans. ADC, který slouží pro nastavení A/D převodníku, a ADC/Repeatedly_trigger_IPBus[Int_TMR], který slouží pro pravidelné přerušení od A/D převodníku. Poté lze napsat v kódu přerušení (Events.c) od A/D převodníku označeného jako funkce `void AD1_OnEnd(void):`

```
AD1_GetValue8(&hodnota);
memory[pocitadlo]=hodnota;
pocitadlo++;
if(pocitadlo>=1100){
    odeslat=1;
    AD1_Disable();}
```

Tento kód se bude provádět vždy v daném časové intervalu. Pro účely této aplikace byl zvolen interval 6,8 μ s. Dochází vždy k načtení 1100 vzorků do paměti mikropočítače, a poté se A/D převodník vypne a umožní mikropočítači provádět další operace. Proměnná pocitadlo počítá počet nasnímaných vzorků, hodnota jako pomocná proměnná pro načítání hodnot s A/D převodníku do paměti, pole memory[] slouží k ukládání vzorkovaných hodnot a proměnná odeslat je určená pro kontrolu, kdy má dojít k odeslání informací v hlavním programu. AD1_GetValue8() je vnitřní funkcí Processor Expert, která načítá poslední převedenou hodnotu s A/D převodníku.

<input checked="" type="checkbox"/>	Bean name	AD1	
<input checked="" type="checkbox"/>	A/D converter	ADCA	ADCA
<input checked="" type="checkbox"/>	Sharing	Disabled	
<input checked="" type="checkbox"/>	Interrupt service/event	Enabled	
<input checked="" type="checkbox"/>	A/D channels	1	
<input checked="" type="checkbox"/>	Channel0		
<input checked="" type="checkbox"/>	A/D channel (pin)	ANA0	ANA0
<input checked="" type="checkbox"/>	A/D channel (pin) signal		
<input checked="" type="checkbox"/>	Mode select	Single Ended	
<input checked="" type="checkbox"/>	Queue	Enabled	
<input checked="" type="checkbox"/>	A/D resolution	Autoselect	12 bits
<input checked="" type="checkbox"/>	Conversion time	6.800 μ s	high: 6.800 μ s
<input checked="" type="checkbox"/>	Trigger configuration wizard	Click to run configurater >	
<input checked="" type="checkbox"/>	Internal trigger	Enabled	
<input checked="" type="checkbox"/>	Source bean	TMR1	
<input checked="" type="checkbox"/>	Sync from PWM	no	
<input checked="" type="checkbox"/>	Number of conversions	1	
<input checked="" type="checkbox"/>	Initialization		
<input checked="" type="checkbox"/>	Enabled in init. code	yes	
<input checked="" type="checkbox"/>	Events enabled in init.	yes	
<input checked="" type="checkbox"/>	CPU clock/speed selection		
<input checked="" type="checkbox"/>	High speed mode	This bean enabled	This bean is enabled
<input checked="" type="checkbox"/>	Low speed mode	This bean disabled	This bean is disabled
<input checked="" type="checkbox"/>	Slow speed mode	This bean disabled	This bean is disabled

Obr. 4.2 Nastavení A/D převodníku

4.2 Zajištění stability a posunu v obraze

Aby bylo zajištěno snímání obrazu od začátku snímku, je potřeba nejdříve nalézt začátek půl snímku. To je možno zajistit nalezení určitého znaku na začátku půl snímku. Tímto znakem je několik impulsů v rozsahu od zatemňovací úrovně po úroveň stejnosměrné složky signálu. Pro tyto účely je nutno rozšířit kód přerušování A/D převodníku o režim testování, ve kterém se načítá vždy pouze několik hodnot a ty jsou v hlavním programu poté kontrolovány a případně, že splní požadované podmínky je spuštěn časovač s periodou 800ms označený jako TI1. Funkce void TI1_OnInterrupt(void) v Events.c.

✓	Bean name	TI1	
✓	Periodic interrupt source	TMRA0123_Compare	TMRA0123_Compare
✓	Counter	TMRA0123	TMRA0123
☐	Interrupt service/event	Enabled	
✓	Interrupt	INT_TMRA3	INT_TMRA3
✓	Interrupt priority	medium priority	3
✓	Interrupt preserve registers	yes	
✓	Interrupt period	800 ms	... high: 800 ms
✓	Same period in modes	yes	
✓	Bean uses entire timer	no	
☐	Initialization		
✓	Enabled in init. code	yes	
✓	Events enabled in init.	yes	
☐	CPU clock/speed selection		
✓	High speed mode	This bean enabled	This bean is enabled
✓	Low speed mode	This bean disabled	This bean is disabled
✓	Slow speed mode	This bean disabled	This bean is disabled

Obr. 4.3 Nastavení časovače TI1

Pro zajištění pravidelného začátku, a tím také snímání určité části obrazu, je nutno využití jednoho časovače. Ten je nutno nastavit na 40ms, jelikož signál má periodu jednoho snímku. Dále je nutno zajistit posunu jednotlivých snímaných prvků, a k tomu je možno využít, například jedné zpožďovací smyčky for s proměnou délkou vykonávání. Poté lze napsat ve vykonávání přerušení od časovače následovný kód:

```
if(odeslat==0) {
    for(x=0;x<zpomal;x++) {
        asm{
            nop}
        zpomal++;
        pocitadlo=0;
        AD1_EnableIntTrigger();
        AD1_Enable();}
}
```

✓	Bean name	TI2	
✓	Periodic interrupt source	TMRB0123_Compare	TMRB0123_Compare
✓	Counter	TMRB0123	TMRB0123
☐	Interrupt service/event	Enabled	
✓	Interrupt	INT_TMRB3	INT_TMRB3
✓	Interrupt priority	high priority	4
✓	Interrupt preserve registers	yes	
✓	Interrupt period	40 ms	... high: 40 ms
✓	Same period in modes	yes	
✓	Bean uses entire timer	no	
☐	Initialization		
✓	Enabled in init. code	yes	
✓	Events enabled in init.	yes	
☐	CPU clock/speed selection		
✓	High speed mode	This bean enabled	This bean is enabled
✓	Low speed mode	This bean disabled	This bean is disabled
✓	Slow speed mode	This bean disabled	This bean is disabled

Obr. 4.4 Nastavení časovače TI2

Přerušení od časovače zůstává zapnuto pro zajištění snímání stejných bodů v obraze. Z tohoto důvodu je nutno nejdříve kontrolovat zda nedochází k jiné činnosti v mikropočítači, jako je například odesílání dat do počítače. Pro tuto kontrolu je určena proměnná odeslat. Poté se provádí zpoždovací smyčka, zajišťující snímání vždy daných bodů v obraze. `AD1_EnableIntTrigger()` a `AD1_Enable()` jsou vnitřní funkce Processor Expertu spouštějící A/D převodník.

4.3 SCI komunikace v EvbDSP56800

V nástroji Processor Expert bylo nutno přidat bean AsynchroSerial, který je propojen s rozhraním SCI. Musí se nastavit požadovaný kanál, s kterým tento bean bude propojen. V rámci tohoto zařízení je nutno zvolit kanál SCI1). Poté je nutno zvolit příslušný Baud rate, ten byl zvolen na 825000 baud. Před odesíláním dat je nutno nejdříve nalézt práh, podle něhož bude docházet k prahování. Tato procedura se musí provádět vždy jednou pro celý nasnímaný obraz.

✓ Bean name	AD1	
✓ A/D converter	ADCA	ADCA
✓ Sharing	Disabled	
⊕ Interrupt service/event	Enabled	
⊖ A/D channels	1	
⊖ Channel0		
✓ A/D channel (pin)	ANA0	ANA0
✓ A/D channel (pin) signal		
⊕ Mode select	Single Ended	
⊕ Queue	Enabled	
✓ A/D resolution	Autoselect	12 bits
✓ Conversion time	6.800 μs	high: 6.800 μs
✓ Trigger configuration wizard	Click to run configurator >	
⊖ Internal trigger	Enabled	
> Source bean	TMR1	
⊕ Sync from PWM	no	
✓ Number of conversions	1	
⊖ Initialization		
✓ Enabled in init. code	yes	
✓ Events enabled in init.	yes	
⊖ CPU clock/speed selection		
✓ High speed mode	This bean enabled	This bean is enabled
✓ Low speed mode	This bean disabled	This bean is disabled
✓ Slow speed mode	This bean disabled	This bean is disabled

Obr. 4.5 Nastavení SCI

```
if (odeslat==1) {
    for (x=0; x<1100; x++) {
        while (AS1_SendChar (memory[x])) {}
    }
    odeslat=0; }
```

Tento kód je v rámci hlavního programu a slouží pro odesílání dat přes sériové rozhraní. Pokaždé, když je nastaven příznak odeslat (kapitole 4.1), dojde pomocí příkazu for k odeslání dat z proměnné `memory[]`, do které se ukládají hodnoty z A/D převodníku. Funkce `AS1_SendChar()` je vnitřní funkcí Processor Expert a slouží k samotnému odesílání dat prostřednictvím SCI rozhraní.

Pro vyhledávání je nutno tento kód rozšířit:

```
if (zpomal==1) {
    max=0;
    min=255;
    for (x=0; x<1100; x++) {
        if (max<memory[x]) {
            max=memory[x];
        }
        if ((min>memory[x]) && (memory[x]>33)) {
            min=memory[x];
        }
    }
    T=(min+max)/2;
}
```

Proměnná zpomal určuje začátek snímku. Hodnoty min a max představují minimální a maximální hodnotu vyskytující se ve snímku. T je hodnota prahu. Po nalezení nového prahu lze rozšířit kód pro odesílání.

```
for (x=0; x<1100; x++) {
    if (T<memory[x]) {
        while (AS1_SendChar(0)) {}
    }
    else {
        while (AS1_SendChar(1)) {}
        while (AS1_SendChar(memory[x])) {}
    }
    odeslat=0;
}
```

4.4 Příjem dat v počítači

Data jsou do počítače přeposílané prostřednictvím USB modulu s čipem FT232RL a je využito ovladače VCP, který vytváří virtuální COM port. Data jsou poté přijímány s využitím COM portu přímo do programu, kde je plněno jednorozměrné pole typu byte (buf2[]).

```
if (!prijem) {
    while ((sP_pri.IsOpen) && (sP_pri.ReadBufferSize != 0)) {
        buf2[velikost] = (byte)sP_pri.ReadByte();
        velikost++;
        if (velikost > 8800) {
            velikost = 0;
            prijem = true;
        }
    }
}
```

Prostřednictvím smyčky while jsou postupně načteny všechny příchozí data s COM portu do buf2[]. Proměnná velikost určuje pozici přijímaných dat v poli buf2 a teprve když dojde k načtení kompletního signálu je nastaven příznak prijem, který souží pro následné zpracovávání a vykreslení informace prostřednictvím časovače.

5 Zpracování obrazové videosekvence

Zpracování obrazu je velice důležitým aspektem při analýze obrazu. Na snímek totiž působí mnoho vnějších vlivů, jako je například špatné nasvícení scény, protisvětlo či velké kontrasty stínů. Pro účely této bakalářské práce je nutno vytvořit program schopný provádět některé jasové transformace, jako je například prahování, popsané v podkapitole 5.1.4 Prahování. Také je nutno si ujasnit pojem histogram a jeho využití při zpracovávání obrazu. Histogram je popsán v podkapitole 5.2 Histogram.

Změnu vzhledu obrazu lze dosáhnout transformací původních jasových úrovní na nové. Zpracovávání obrazu lze provádět v prostorové doméně či frekvenční doméně. Pod pojmem zpracování obrazu v prostorové doméně je definovaná práce s funkcí $f(x,y)$, která je brána jako diskrétní funkce jednotlivých jasových úrovní, s definičním oborem hodnot daným velikostí zpracovávaného obrazu $\langle x_{\min}, x_{\max} \rangle \times \langle y_{\min}, y_{\max} \rangle$. Naopak ve frekvenční doméně jde o zpracování obrazu na základě metod aplikovaných na Fourierovu transformaci původního obrazu.

[2]

5.1 Transformace jasu obrazu

Původní obraz s jasovými hodnotami je možno označovat jako funkci $f(x,y)$, která bude transformována do nových jasových hodnot označených funkcí $g(x,y)$. Pak je tedy možno zjednodušeně zapsat do tvaru:

$$g(x, y) = T\{f(x, y)\} \quad 5.1$$

kde $f(x,y)$ je původní obraz
 $g(x,y)$ je výstupní obraz
 T má význam libovolné transformace jasu

Jedny z nejjednodušších jasových transformací nevyužívají okolí právě zpracovávaného bodu, ale pouze jeho jasovou hodnotu. Mezi takové transformace patří, například prahování, negativ, roztažení kontrastu či Gama korekce. Naopak složitější transformace jsou založeny na zjišťování okolí daného bodu, jako je například konvoluce nebo jiné nelineární operace, které se využívají pro ostření či vyhlazování obrazu.

Pro zjednodušování transformací lze uvažovat, že daná jasová hodnota je z intervalu $\langle 0,1 \rangle$, kde 0 je považována jako hodnota absolutní černé a 1 je považována za absolutní bílou barvu (ovšem je možno využívat i opačnou konvenci). Skutečné hodnoty se ovšem pohybují v intervalu $\langle 0, L-1 \rangle$, kde L je maximální hodnota jasu, která se může vyskytnout v obraze. Například pro obraz s 256 hodnotami jasu (8 bitová hodnota) je L rovna 256, nebo-li 2^8 .

[2]

Jako obrázek pro ukázání funkce jednotlivých transformací byl použit obrázek zachycený s pomocí zařízení vytvořeného pro tuto bakalářskou práci.



Obr. 5.1 Původní obrázek zachycený kamerou

5.1.1 Transformace obrazu na negativ

Negativ je jedna s nejjednodušších transformací. Historicky negativ vznikl pro převod ze starých filmů fotoaparátů a následným vyvoláním negativu vznikl původní obraz. Ve své podstatě se jedná o nahrazení jasové hodnoty hodnotou opačnou což můžeme přepsat do vzorce:

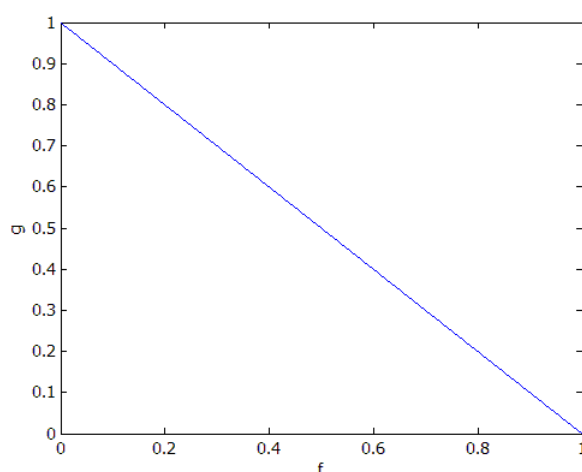
$$g = 1 - f \quad 5.2$$

pro f v intervalu $\langle 0,1 \rangle$, pokud předpokládáme že funkce f je v intervalu $\langle 0,L-1 \rangle$ pak můžeme přepsat vzorec na tvar:

$$g = (L-1) - f \quad 5.3$$

na Obr. 5.2 je možno vidět transformační křivku negativu.

[2]



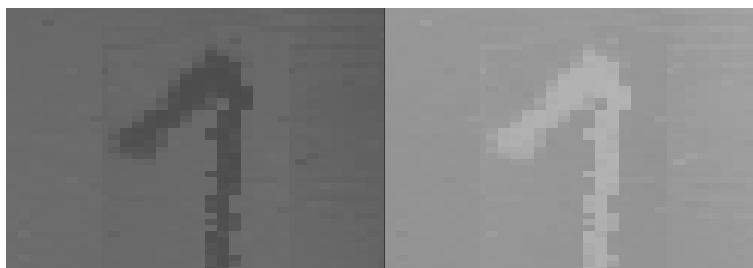
Obr. 5.2 Transformační křivka negativu [2]

Pro převod obrazu na negativ je použit velice jednoduchý algoritmus. Objekt `flag_BW_p[,]` je dvojrozměrné pole datových typů `int` představující původní obraz, `flag_neg` je pro změnu objekt typu `bitmap`, do něj se vytváří negativ původního obrazu.

```

for (x = 0; x < flag_puvod.Width; x++)
{
    for (y = 0; y < flag_puvod.Height; y++)
    {
        //puvodni bitmapa prevedena na negativ
        flag_neg.SetPixel(x,y,Color.FromArgb(255-flag_BW_p[x,y],Color.Black));
    }
}

```



Obr. 5.3 Vlevo původní obraz, vpravo obraz po negaci

5.1.2 Gama korekce obrazu

Historickým důvodem používání Gama korekce byla nelinearita závislosti televizních obrazovek. Klasická obrazovka se žhavenou katodou označována také jako CRT obrazovky (Cathode Ray Tube) neměli lineární převod mezi napětím, které představovalo jistou úroveň šedi, a výsledným jasnem obrazovky, ale spíše se dala aproximovat křivkou vyjádřenou rovnicí:

$$g = C \cdot (f - \varepsilon)^\gamma \quad 5.4$$

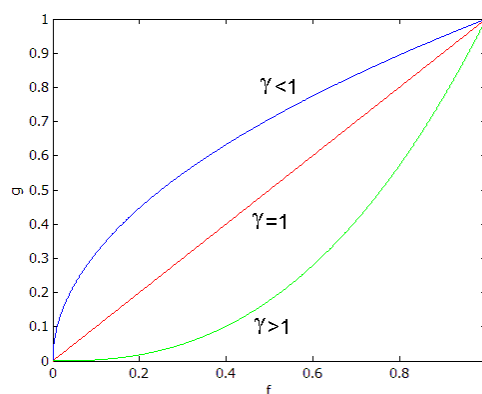
Kde - C je konstanta vynásobení

- ε je posun křivky

- γ je konstanta pohybující se z pravidla mezi 1,8 a 2,4

Ovšem ε tedy posun křivky (tzv. offset) se většinou nepoužívá, proto lze tuto rovnici přepsat do tvaru:

$$g = C \cdot f^\gamma \quad 5.5$$



Obr. 5.4 Transformační křivka Gama korekce [2]

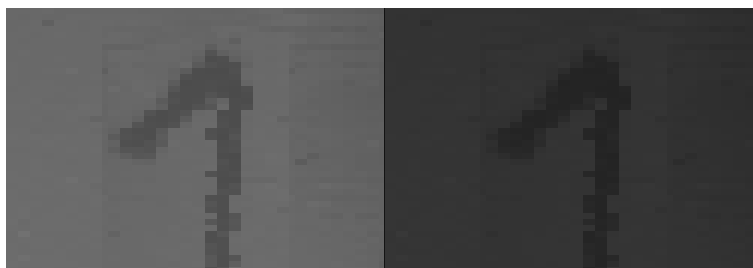
Gama korekce se nepoužívá pouze při úpravě obrazu, ale také se dá použít v případě špatné zvolené expozice, kdy je část objektu skryta ve stínu.

[2]

Při znalosti těchto faktů lze vytvořit jednoduchý algoritmus pro výpočet gama korekce:

```
for (x = 0; x < flag_puvod.Width; x++)
{
    for (y = 0; y < flag_puvod.Height; y++)
    {
        //samotna gama korekce
        intensita = flag_BW_p[x, y];
        //pocitani nove intenzity pomoci gama korekce
        intensita=(int) (Math.Min(255, (int) ((Math.Pow((intensita/255f), (1f/koef
        ))*255))));
        //vytvareni bitmapy po gama korekci
        flag_g_kor.SetPixel(x,y,Color.FromArgb((intensita),Color.Black));
    }
}
```

Kde *intensita* představuje novou intenzitu vypočtenou pomocí gama korekce, *koef* představuje γ z výše uvedené rovnice, *flag_g_kor* představuje novou bitmapu po provedení gama korekce.



Obr. 5.5 Vlevo původní obraz, vpravo obraz po gama korekci

5.1.3 Roztažení kontrastu obrazu

Roztažení kontrastu se většinou používá v případech, kdy dochází k příliš malým rozdílům mezi maximální a minimální hodnotě jasu v obraze. Pro tuto metodu je ovšem nutno nejdříve nalézt tzv. globální maximum a minimum vyskytující se v obraze. Poté lze říci, že jasové hodnoty v obraze se pohybují v intervalu $\langle f_1, f_2 \rangle$. U roztažení kontrastu se ve své podstatě jedná o roztáhnutí tohoto intervalu na interval kompletní, tedy $\langle 0,1 \rangle$. Toho lze zajistit pomocí vzorce:

$$g = \frac{f - f_1}{f_2 - f_1} \quad 5.6$$

Kde - f_1 je globální minimum

- f_2 je globální maximum

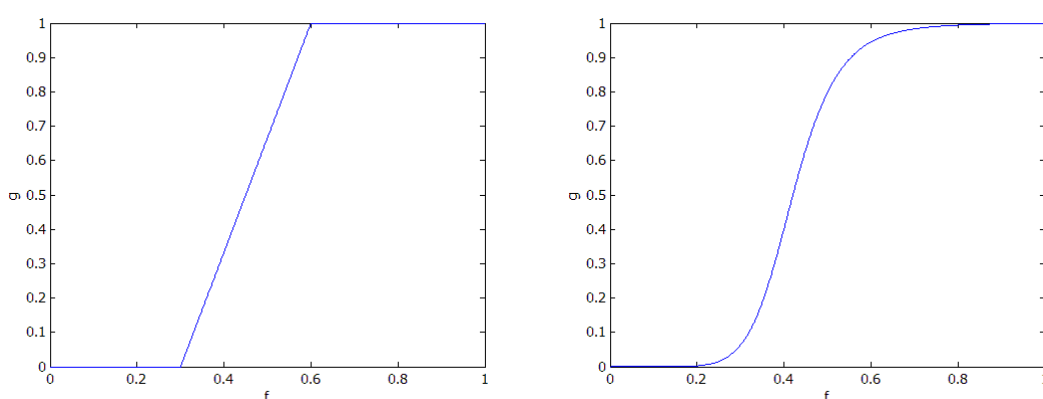
Ovšem v tomto případě dochází k lineárnímu roztažení kontrastu transformační křivky, což lze vidět na Obr. 5.6 (vlevo).

Občas se využívá namísto lineárního rozložení kontrastu také nelineární (Obr. 5.6 vpravo). V těchto případech lze tuto transformaci popsat vzorcem:

$$g = \frac{1}{1 + \left(\frac{m}{f + \varepsilon} \right)^E} \quad 5.7$$

Kde

- m určuje střed roztažení
- E určuje strmost křivky
- ε je konstanta zabráňující dělení nulou



Obr. 5.6 Transformační křivka roztažení kontrastu [2]

Za další způsoby roztahování kontrastu se také může považovat tzv. prahování. U prahování se jedná o převedení všech úrovní šedi na dvě hodnoty, a to na hodnotu černé a bílé barvy podle meze prahování T , která vyznačuje hodnotu šedi od níž bude hodnota bílá nebo černá. Tedy přesněji pro interval $\langle 0, T \rangle$ vstupního obrazu (f) je výstupní hodnota $g = 0$ a pro interval $\langle T, 1 \rangle$ je $g = 1$.

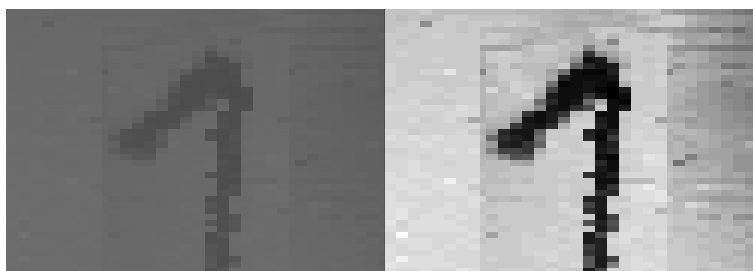
[2]

Nejdříve je nutno nalézt minimální a maximální hodnotu jasu v obraze, a to lze provést takto:

```
for (x = 0; x < flag_puvod.Width; x++)
{
    for (y = 0; y < flag_puvod.Height; y++)
    {
        pomocna = (float)((float)(flag_BW_p[x, y])/255);
        if (pomocna < minjas)
            minjas = pomocna;
        if (pomocna > maxjas)
            maxjas = pomocna;
    }
}
```

Po nalezení maximální (maxjas) a minimální (minjas) hodnoty jasu lze již snadno přepočítat všechny jasové hodnoty v obraze pomocí algoritmu, ve kterém koef představuje novou hodnotu jasu v intervalu $\langle 0,1 \rangle$, ten je přepočítán na skutečnou intenzitu (intensita) vynásobením maximální hodnotou jasu. Poté flag_roz_kon objekt typu bitmap představující obraz po metodě roztažení kontrastu.

```
for (x = 0; x < flag_puvod.Width; x++)
{
    for (y = 0; y < flag_puvod.Height; y++)
    {
        pomocna=(float) ((float) (flag_BW_p[x,y]) /255);
        koef=(pomocna-minjas) / (maxjas-minjas);
        intensita=(int) (koef*(255f));
        flag_roz_kon.SetPixel(x,y,Color.FromArgb(intensita,Color.Black));
    }
}
```



Obr. 5.7 Vlevo původní obraz, vpravo obraz po roztažení kontrastu

5.1.4 Prahování obrazu

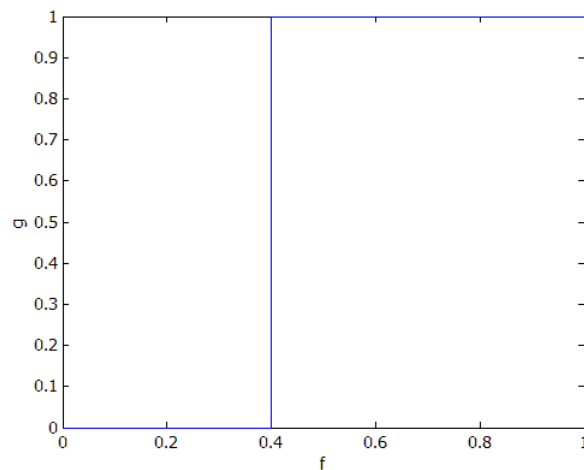
Prahování je jednoduchý způsob filtrace obrazu. Jedná se o jednoduché převedení s více úrovní šedi na dvě úrovně, a to bílou a černou. Používá se v případech vyhledávání určitých tvarů v obraze.

Ve své podstatě se provádí pouze porovnávání dané hodnoty určitého pixel s hodnotou prahu T . Jestliže je hodnota jasu větší nebo rovna hodnotě prahu, je možné výslednou hodnotu v prahovaném obraze definovat jako rovnou jedné (bíle). Pokud je menší lze uvažovat, že hodnota je rovna nule (černé). Touto metodou lze obraz rozdělit na popředí (objekt zájmu) a pozadí. (Obr. 5.8)

Prahování lze rozdělit na globální a lokální. U globálního prahování se hledá nebo je zvolena pouze jedna hodnota prahu, která se využívá pro celý obraz. Na rozdíl od lokálního prahování, kde dochází ke změně pro danou část obrazu.

Hodnotu prahu je možno zvolit různými způsoby, například zvolit na základě maxima a minima nebo pomocí histogramu obrazu, popřípadě ji může zvolit také uživatel.

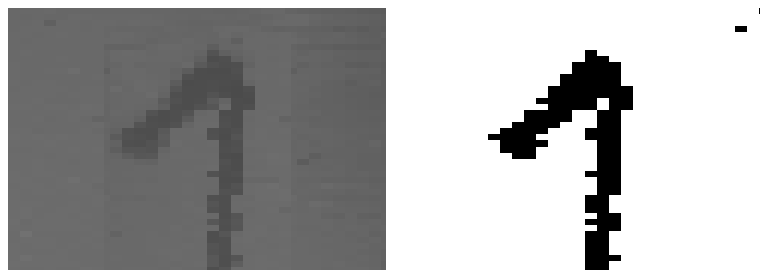
[2]



Obr. 5.8 Transformační křivka prahování [2]

```
for (x = 0; x < flag_puvod.Width; x++)
{
    for (y = 0; y < flag_puvod.Height; y++)
    {
        if (flag_BW_p[x, y] > T)
        {
            flag_prah.SetPixel(x, y, Color.Black);
        }
        else
        {
            flag_prah.SetPixel(x, y, Color.White);
        }
    }
}
```

Algoritmus pro jednoduché prahování s využitím předem dané hodnoty je velice jednoduchý. T představuje samotnou hodnotu prahu, podle kterého bude provedeno prahování, flag_prah totéž představuje nový obraz po provedení prahování.



Obr. 5.9 Vlevo původní obraz, vpravo obraz po prahování

5.2 Histogram obrazu

Histogram znázorňuje rozložení jasových hodnot v obraze. Přesněji jde o grafické znázornění četnosti jednotlivých úrovní jasu. S toho vyplývá, že rovnice pro výpočet je tedy diskrétní funkce:

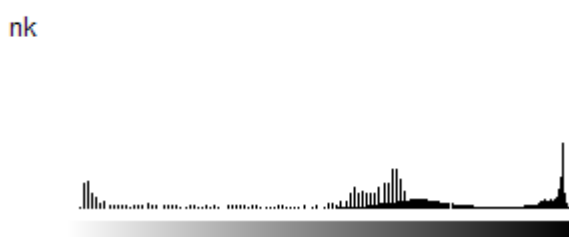
$$h(f_k) = n_k \quad 5.8$$

Kde - f_k je příslušná hodnota jasu

- h_k je četnost f_k

- k nabývá hodnot $0, 1, \dots, L-1$

V případě barevného obrazu se histogram rozděluje do čtyř grafů, které znázorňují jednotlivé složky barev a jasová hodnota celkového obrazu. Na Obr. 5.10 je možno vidět vykreslený jasový histogram obrazu. (osa X představuje jednotlivé úrovně jasu (f_k) a osa Y představuje jednotlivé četnosti dané úrovně šedi)



Obr. 5.10 Jasový histogram obrazu

Histogram slouží pro některé druhy filtrací, např. u filtrací založených na vyrovnávání histogramu nebo hledání optimálního prahu při prahování.

Pro vykreslování se histogram musí nejdříve normalizovat vydělením jednotlivých jasových úrovní celkovým počtem bodů v obraze (pixel), pro tento případ lze využít vzorec:

$$p(f_k) = \frac{n_k}{N} \quad 5.9$$

Kde - p_k je pravděpodobnost jasu

- N je celkový počet bodů v obraze

čímž získáme odhad pravděpodobnosti výskytu jednotlivých úrovní jasu v obraze.

[2]

Díky těmto znalostem je možno napsat jednoduchý postup, kde histo_puvod představuje pole hodnot histogramu. Poté se již musí pouze projít celý obraz a postupně vyhodnocovat intenzitu jednotlivých pixelů.

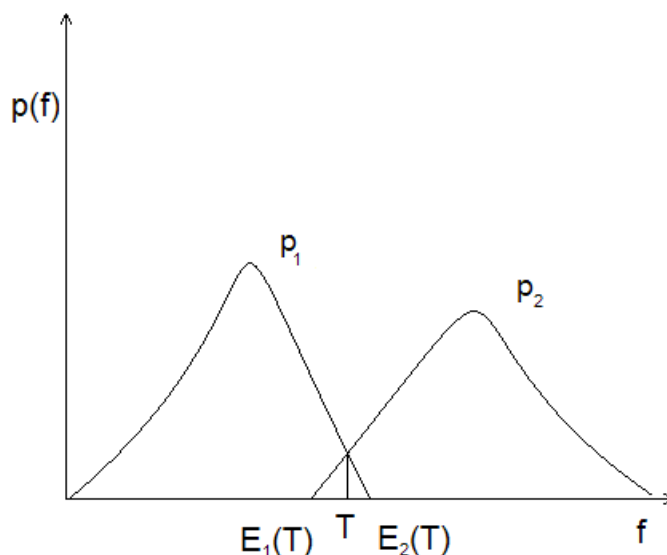
```
for (x = 0; x < flag_puvod.Width; x++)
{
    for (y = 0; y < flag_puvod.Height; y++)
    {
        histo_puvod[flag_BW_p[x, y]]++;
    }
}
```

6 Analýza videosekvence

Principů vyhledávání je v praxi používáno mnoho, některé jsou založeny dokonce i na umělé inteligenci. Principy používané v této bakalářské práci nejsou založeny na UI, ale pouze na porovnávání objektu s jiným. Pro tyto principy je nutno nejdříve rozdělit obraz na dvě úrovně, a to bílou a černou barvu. K tomu bylo využito prahování popsáno v kapitole 5 Zpracování obraz. Z důvodu nutnosti nalézt vhodný práh pro převod odstínu šedi na čistě bílou a černou barvu, je nutno nejdříve provést metody automatického nalezení prahu. Popis metody, která byla pro tento účel vybrána je v podkapitole 6.1. V této podkapitole je také popsán kód, který byl odvozen s této metody pro účel této bakalářské práce. Další činností, kterou je nutno provést před samotným vyhledáváním, je vymezení míst, ve kterých se s pravděpodobností nalézají příslušné znaky.

6.1 Metoda automatické hledání prahu

Je mnoho různých metod pro automatické hledání prahu, ale v zásadě se jedná vždy o nalezení takové hodnoty prahu, při které bude nejlépe zajištěno oddělení popředí (objekt zájmu) a pozadí. Ovšem tyto metody jsou vždy založeny na předpokladu dvou jasných vrcholů viditelných na histogramu.



Obr. 6.1 Hledání optimálního prahu

Například lze použít Otsuovu metodu, která předpokládá, že lze histogram aproximovat dvěma křivkami reprezentujícími objekt zájmu a pozadí. Tato metoda se snaží minimalizovat chybu mylného zařazení popředí jako pozadí a naopak, která závisí právě na správném zvolení prahu. Histogram na Obr. 6.1 ukazuje právě takové rozložení dvou křivek p_1 a p_2 , kde $E_1(T)$ reprezentuje chybu klasifikace popředí a ta lze vypočítat pomocí vzorce:

$$E_1(T) = \int_{-\infty}^T p_2(f) df \quad 6.1$$

opačná chyba označená jako $E_2(T)$, která naznačuje chybnou klasifikaci pozadí lze napsat ve tvaru:

$$E_2(T) = \int_T^{\infty} p_1(f) df \quad 6.2$$

Jelikož je nepravděpodobné, že by popředí a pozadí zabíralo stejnou plochu. Je nutno zavést pravděpodobnost výskytu popředí (P_1) a pozadí (P_2), přičemž platí pravidlo $P_1 + P_2 = 1$. Poté pro získání celkové chyby $E(T)$ lze využít rovnici:

$$E(T) = P_2 E_1(T) + P_1 E_2(T) \quad 6.3$$

U této metody je snaha o výběr takového prahu, při kterém tato celková chyba bude minimální.

Tuto metodu je možno zjednodušit. V závislosti na zvoleném prahu lze normalizovaný histogram rozdělit na dvě nové množiny bodů. Jedna množina by měla představovat pozadí (C_0) a je charakterizována sumami hodnot μ_0, ε_0 .

$$\mu_0 = \sum_{k=0}^{T-1} p(k) \quad 6.4$$

$$\varepsilon_0 = \sum_{k=0}^{T-1} \frac{k \cdot p(k)}{\mu_0} \quad 6.5$$

Druhá množina poté představuje popředí (C_1) a je ji možno charakterizovat μ_1, ε_1 .

$$\mu_1 = \sum_{k=T}^{L-1} p(k) \quad 6.6$$

$$\varepsilon_1 = \sum_{k=T}^{L-1} \frac{k \cdot p(k)}{\mu_1} \quad 6.7$$

Dále je potřeba střední hodnota v celém histogramu.

$$\varepsilon_T = \sum_{k=0}^{L-1} k \cdot p(k) \quad 6.8$$

Poté je možno nalézt optimální prach jako maximální hodnota následujícího vzorce:

$$\delta^2 = \mu_0 (\varepsilon_0 - \varepsilon_T)^2 + \mu_1 (\varepsilon_1 - \varepsilon_T)^2 \quad 6.9$$

[2]

Tyto znalosti je možné využít při sestavení programu pro automatické vyhledání hodnoty prahu. Nejdříve je nutno normalizovat histogram (his_p) to je možné pomocí podělení hodnoty histogramu celkovou plochou obrazu.

```
plocha = flag_puvod.Width * flag_puvod.Height;
for (x = 0; x < 256; x++)
{
    his_p[x] = (float)(his[x]) / (float)(plocha);
}
```

Poté je nutno vypočítat nejdříve hodnotu ε_T (fit), která zůstává konstantní po celou dobu vyhledávání prahu.

```
fit = 0;
for (x = 0; x < 256; x++)
{
    fit = fit + x * his_p[x];
}
```

Aby bylo možno určit správně novou hodnotu prahu, je nutno vypočítat hodnotu δ (om[]) pro všechny urovně jasu. To lze vypočítat takto ($om0 - \mu_0$, $fi0 - \varepsilon_0$, $om1 - \mu_1$, $fi1 - \varepsilon_1$):

```
for (x = 0; x < 256; x++)
{
    om0 = 0;
    for (y = 0; y < x; y++)
    {
        om0 = om0 + his_p[y];
    }

    fi0 = 0;
    for (y = 0; y < x; y++)
    {
        fi0 = fi0 + (y * his_p[y] / om0);
    }

    om1 = 0;
    for (y = x; y < 256; y++)
    {
        om1 = om1 + his_p[y];
    }

    fi1 = 0;
    for (y = x; y < 256; y++)
    {
        fi1 = fi1 + (y * his_p[y] / om1);
    }
    om[x] = (om0 * (fi0 - fit) * (fi0 - fit) + om1 * (fi1 - fit) *
(fil - fit)) * (om0 * (fi0 - fit) * (fi0 - fit) + om1 * (fi1 - fit) *
(fil - fit));
}
```

Základní smyčka slouží pro výpočet hodnoty δ pro všechny jasové hodnoty. Poté je již velice snadné nalézt hodnotu prahu. Pouze je nutno projít celé pole představující všechny hodnoty δ v celém histogramu a vybrat tu nejvyšší. (max - maximální hodnota δ , x je příslušná hodnota jasu)

```

for (x = 0; x < 256; x++)
{
    if (max < om[x])
    {
        T = x;
        max = om[x];
    }
}

```

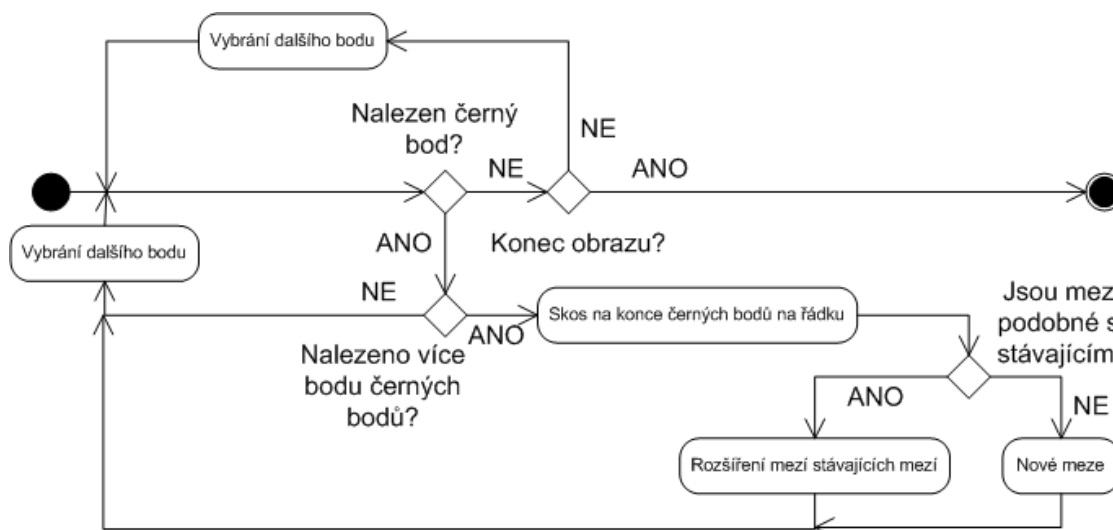
6.2 Detekce spojitostí v obraze

Hlavním problémem většiny metod využívaných při hledání objektů v obraze, především znaků, je závislá na posunutí objektu v obraze. Tato závislost má za následek nesprávnou identifikaci objektu. Z tohoto důvodu je výhodnější nejdříve nalézt objekt zájmu a provést jeho ohrazení.

[2]

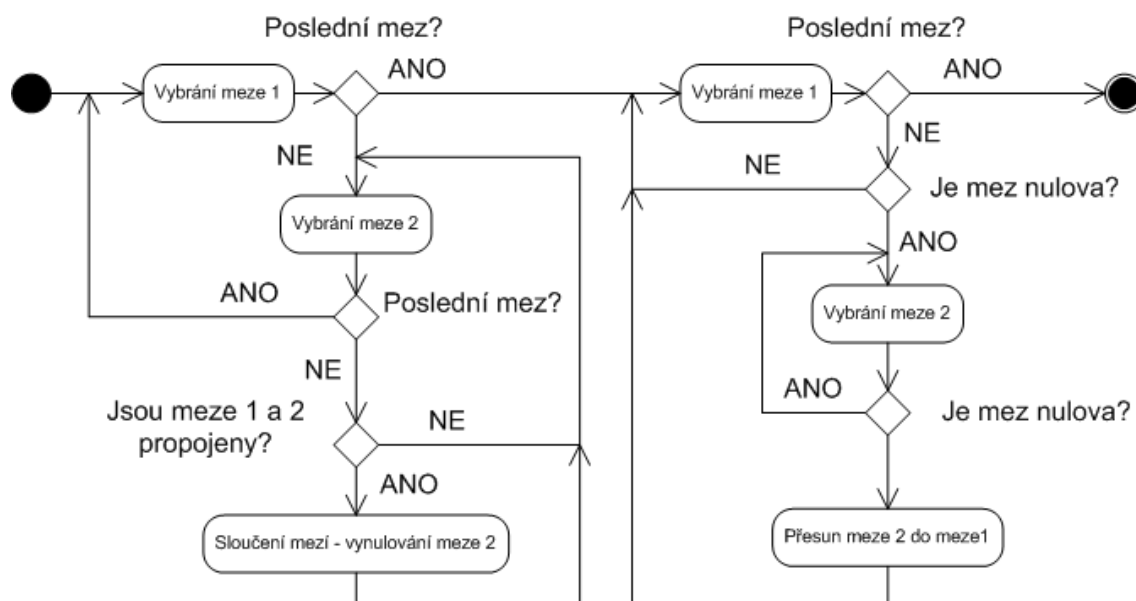
Existují různé metody, jak nalézt hledaný objekt v obraze a ohrazení jej. Toto ohrazení je možno nazvat meze objektu a představují umístění objektu v obraze. Hledání spojitosti je metoda, která je schopna vyhledávat pouze spojitě objekty. Její výhodou je možnost nalezení více objektů skrývajících se v obraze.

Hledání spojitosti v obraze je založeno na prohledání celého obrazu řádek po řádku a nalezení černého bodu. V případě, že dojde k nalezení černého bodu je nutno zjistit, zda se nejedná pouze o anomálii vyskytující se v obraze. To lze zjistit například kontrolou vedlejšího bodu. Pokud se nejedná o samostatný bod je nutno uložit polohu tohoto bodu a zjistit poslední bod s hodnotou černé barvy v řadě. Po nalezení posledního bodu v řadě je zapotřebí zkontrolovat všechny předchozí meze, které již byly nalezeny, s tímto nálezem. V případě, že dojde k nálezů mezí, které jsou souhlasné s umístěním mezí o řádek výše, jsou tyto meze příslušně poupraveny anebo jsou přidány nové meze.



Obr. 6.2 Diagram algoritmu pro vyhledávání spojitostí

Problém může nastat u některých spojitostí. Pokud je totiž spojitost na prvních řádcích rozdělená a spojuje se na dalších, dojde k vytvoření dvou mezí místo jedné a poté je nutno projít všechny meze opět a zjistit jestli nejsou náhodou některé z nich propojeny. Tato metoda nalezení spojitostí je velice kvalitní a dokáže vyhledat všechny spojitosti v obraze.



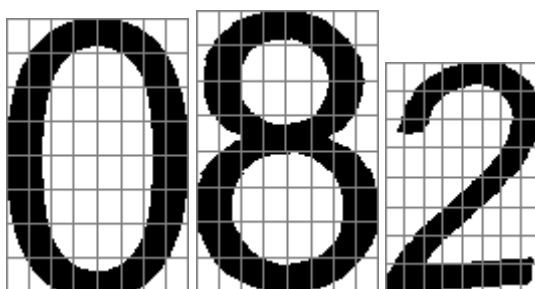
Obr. 6.3 Diagram algoritmu pro úpravu mezí

6.3 Analýza znaků v obraze

Existují různé způsoby vyhledávání znaků či tvarů v obraze. Ovšem dost záleží právě na typu hledaných tvarů. Ve své podstatě se vždy jedná o porovnávací metodu dvou prvků, a to snímaného objektu s určitou předlohou. Problémem je natočení, velikost a posun objektu než je předloha, proto je nutno vytvářet metody, které eliminují jak natočení, tak velikost.

6.3.1 Segmentace objektu

Jedním ze způsobů je segmentace objektu. Tato metoda je velmi účinná při vyhledávání velkých znaků, ale nejdříve je nutno nalézt spojitosti v obraze (podkapitola 6.2). Je založena na rozdělení objektu pomocí mřížky do více segmentů. To je možno vidět na Obr. 6.4.



Obr. 6.4 Příklady segmentovaných objektů

Aby byla zajištěna jistá úroveň normalizace, a tím také odstraněna závislost na velikosti, je nutno vypočítat rozložení mřížky vytvářející plochu o velikosti 8x8 segmentů. Nejdříve je zapotřebí vypočítat matice m_1 a m_2 sloužící jako pomocné matice. Matice m_1 a m_2 musí obsahovat polovinu počtu prvků než je počet segmentů výsledné matice a nulu. Poslední prvek matic musí být poloviční hodnota šířky ($x_{\max} - x_{\min}$) pro matici m_1 a poloviční hodnotu výšky ($y_{\max} - y_{\min}$) pro matici m_2 , další prvky matice se poté dají vypočítat:

$$m_1(n) = \frac{n-1}{seg} (x_{\max} - x_{\min}) \quad 6.10$$

Kde - $m_1(n)$ je n prvek matice
 - n je číslo prvku (nabývá hodnot 1, 2, 3...)
 - seg je počet segmentů

Jednotlivé prvky matic m_1 a m_2 musí být zaokrouhleny na celá čísla. To je dáno diskretizací obrazu.

$$m_1 = \begin{bmatrix} 0 & \frac{1}{8}(x_{\max} - x_{\min}) & \frac{2}{8}(x_{\max} - x_{\min}) & \frac{3}{8}(x_{\max} - x_{\min}) & \frac{4}{8}(x_{\max} - x_{\min}) \end{bmatrix} \quad 6.11$$

$$m_2 = \begin{bmatrix} 0 & \frac{1}{8}(y_{\max} - y_{\min}) & \frac{2}{8}(y_{\max} - y_{\min}) & \frac{3}{8}(y_{\max} - y_{\min}) & \frac{4}{8}(y_{\max} - y_{\min}) \end{bmatrix} \quad 6.12$$

Kde - m_1 je pomocná matice pro osu x
 - m_2 je pomocná matice pro osu y
 - x_{\min} a x_{\max} jsou maximální a minimální hodnoty nalezeného objektu na ose x
 - y_{\min} a y_{\max} jsou maximální a minimální hodnoty nalezeného objektu na ose y

Poté je možno vypočítat rozmezí jednotlivých segmentů pomocí jednoduchého maticového výpočtu jako:

$$s_1 = x_{\min} + m_1 \quad 6.13$$

$$s_2 = x_{\max} - m_1 \quad 6.14$$

$$s_3 = y_{\min} + m_2 \quad 6.15$$

$$s_4 = y_{\max} - m_2 \quad 6.16$$

$$s_x = [s_1 \quad s_2] \quad 6.17$$

$$s_y = [s_3 \quad s_4] \quad 6.18$$

Kde - s_x je matice souřadnic na ose x
 - s_y je matice souřadnic na ose y

Pomocí matic s_x a s_y je možno vytvořit souřadnicovou síť pro každý nalezený objekt a lze vypočítat jasové pokrytí jednotlivých segmentů.

$$sum(n, m) = \sum_{x=s_x(n)}^{s_x(n+1)} \sum_{y=s_y(m)}^{s_y(m+1)} f(x, y) \quad 6.19$$

- Kde
- $sum(n, m)$ je hustota pokrytí příslušného segmentu
 - $f(x, y)$ je jasová hodnota daného prvku (zde nabývá hodnot 0 nebo 1)
 - n a m mohou nabývat hodnot 1 až 9 (pro matici 8x8)

Procentuální pokrytí pro jednotlivé segmenty lze následně vypočítat pomocí vzorce:

$$poc(n, m) = \frac{sum(n, m)}{(s_x(n+1) - s_x(n)) \cdot (s_y(m+1) - s_y(n))} \cdot 100\% \quad 6.20$$

Kde $poc(n, m)$ je procentuální pokrytí černé barvy segmentu n, m , který je omezen souřadnicemi $s_x(n)$ a $s_x(n+1)$ na ose x a $s_y(n)$ a $s_y(n+1)$ na ose y .

Takto je získaná matice poc o velikosti n, m :

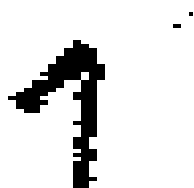
$$poc = \begin{bmatrix} poc(1,1) & \cdots & poc(1,m) \\ \vdots & \ddots & \vdots \\ poc(n,1) & \cdots & poc(n,m) \end{bmatrix} \quad 6.21$$

Tuto matici je možno poté využít pro sestavení již normalizované matice obsahující stejný počet prvků jako matice poc o hodnotě pouze 0 nebo 1. Je nutno určit mezní procentuální pokrytí černou barvou. Tato hodnota musí být optimálně nastavena, zde je mezní procentuální pokrytí zvoleno mez_p na 25%. Poté je možno určit matici poc_n , což je právě normovaná matice. Její prvky jsou definovány pomocí mezního procentuálního pokrytí. Je možno zvolit konvenci, jestliže $poc(n, m)$ je menší nebo rovna mez_p poté je prvek $poc_n(n, m)$ roven 0, pokud je hodnota $poc(n, m)$ větší je než mez_p je možno prvek n, m matice poc_n definovat jako 1. Je ovšem možno použít i opačnou konvenci, ale musí být stejná jako byla použita u předlohy

Tato matice může vypadat, například pro ideální číslo 1, takto:

$$poc_n = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \end{bmatrix} \quad 6.22$$

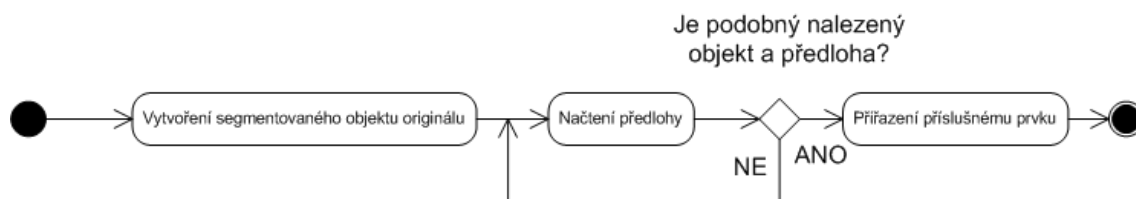
Pro neideální obraz jedničky viditelný na Obr. 6.5 je tímto způsobem možno vytvořit matici 6.22.



Obr. 6.5 Neideální obraz jedničky

$$poc_n = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 \end{bmatrix} \quad 6.23$$

Nyní je obraz normalizovaný a na tento normalizovaný tvar již nepůsobí velikost objektu ani natočení. Stačí již jen využít jednoduchou funkci pro porovnání dvou matic a porovnávat jejich odchylku.



Obr. 6.6 Diagram algoritmu porovnávání předlohy a originálu



Obr. 6.7 Předloha pro čísla

Tato metoda má svou výhodu například díky možnosti vytvoření libovolné předlohy pro porovnávání s objektem zájmu skrývajícím se v obraze. Na Obr. 6.7 je možno vidět předlohu vykreslenou pomocí matic předloh pro čísla. Za předpokladu vytvoření předlohy pro písmena tento algoritmus bude schopen tyto písmena také identifikovat.

Také je možno určit procentuální odchylku předlohy od nalezeného objektu $E(p)$ jako počet rozdílných bodů p_r ku celkovému počtu p_c bodů v předloze.

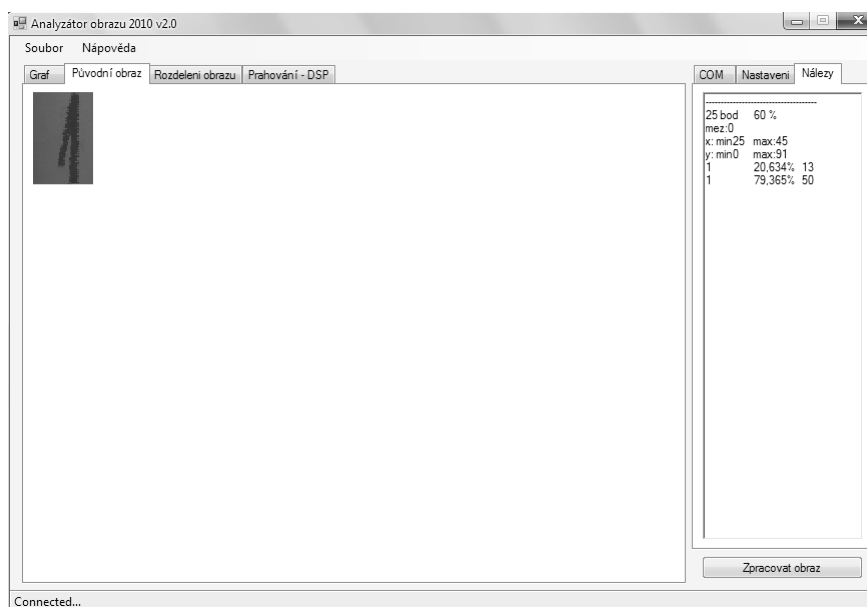
$$E(p) = \frac{p_r}{p_c} \cdot 100\%$$

6.24

Při vyhledávání poté tato odchylka může definovat míru odlišnosti předlohy a nalezeného objektu.

6.4 Uživatelské rozhraní

Pro účely této bakalářské práce byl vytvořen program pro komunikaci se systémem pro monitorování obrazu, vykreslování zachycovaných snímků, jejich ukládání a komplexnější analýzu obrazu. Tento program je možno využít s příslušným zařízením, ale také jej lze využít samostatně a otevřít libovolný obrázek uložený na počítači. Záložky na levé straně programu mají pouze zcela informativní charakter. Především zobrazují jednotlivé obrázky po příslušných transformacích jasu. V případě připojení zařízení dochází k zobrazování obrazu zachycovaného pomocí videokamery v záložce Původní obraz. V záložce Prahování – DSP se zobrazuje obraz, který byl prahován v zařízení a byl odeslán společně s původním obrazem pro zpracování. V záložce graf je spojitě vykresluje digitalizovaný signál, po přijetí jednotlivých částí a jejich složení. Tento Graf se zobrazuje pouze za předpokladu snímání obrazu s využitím zařízení.



Obr. 6.8 Program Analyzátor obrazu 2010

Záložka COM slouží pro připojení k zařízení a nastavení rychlosti komunikace s COM Port. V případě, že zařízení bylo připojeno k počítači až po spuštění programu, je nutno stisknout tlačítko Vyhledat COM Port, a to vyhledá všechny aktuálně připojené porty. Po zadání správného COM Portu a správné rychlosti Baud rate je zapotřebí pro připojení k zařízení stisknout tlačítko Connect. Pro odpojení od zařízení je nutno stisknout tlačítko Disconnect. Po stisku tlačítka Zachytit obraz dojde k zachycení posledního zobrazeného snímku na záložce Původní obraz a je umožněno s tímto obrazem pracovat.

Záložka nastavení slouží pro nastavení programu. Je zde umožněno ručně nastavit koeficient gama korekce. Po zatržení možnosti Zobrazovat Gama Korekci se vlevo objeví záložka Gama korekce, ve které je zobrazen výsledek použití Gama korekce na snímek. Je zde možno také zvolit, zda se bude provádět automatické prahování nebo bude ručně zvolena hodnota prahu. Záložka prahování se bude zobrazovat pouze za předpokladu, že bude zatržena možnost Zobrazovat Prahování. V Nastavení vyhledávání je možno zvolit minimální procento černé barvy, při které se bude uvažovat o černé barvě, ale přesnost při vyhledávání. V případě více předloh pro porovnávání je možno si vybrat příslušnou předlohu v nabídce Předlohy. V Ostatní je možno zvolit další zobrazované záložky Histogram (zobrazí se dvě záložky obsahující histogramy Gama korekce, původního obrazu, negativ a roztažení kontrastu), Negativ a roztažení kontrastu.

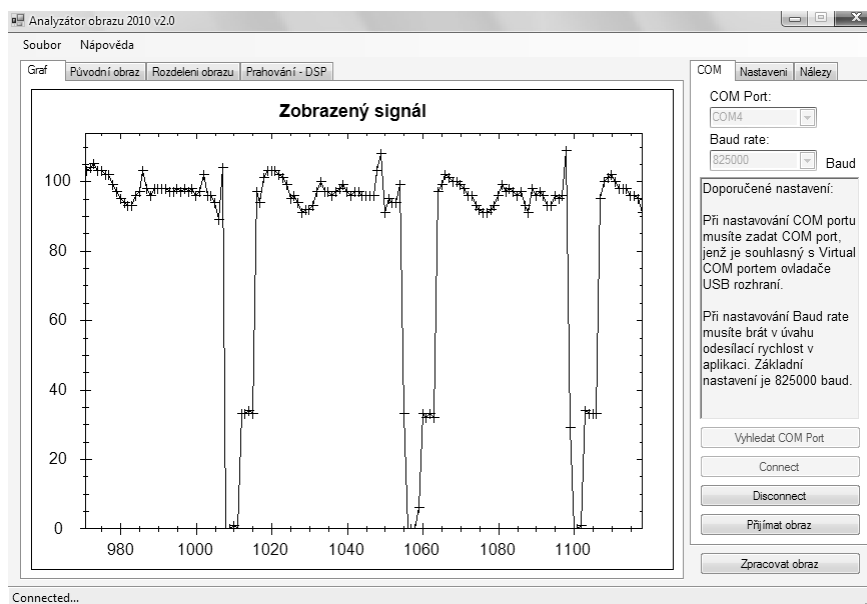
Ve složce nálezy se vypisují v textové formě všechny nalezené objekty s příslušnou odchylkou mezi předlohou a skutečným obrazu.

The screenshot shows the 'Nastavení' (Settings) window with three tabs: 'COM', 'Nastavení', and 'Nálezy'. The 'Nastavení' tab is active, displaying various configuration options. The 'COM' tab shows 'COM Port' set to 'COM4' and 'Baud rate' set to '825000'. The 'Nastavení' tab is divided into several sections: 'Gama korekce' (Gamma correction) with a checkbox 'Zobrazovat Gama Korekci' and a 'Koeficient Gama korekce' set to '1.8'; 'Prahování' (Thresholding) with a checkbox 'Zobrazovat prahování', a checked 'Automatické prahování' checkbox, and a 'Ruční nastavení prahu' set to '125'; 'Nastavení vyhledávání' (Search settings) with 'max. počet špatných bodu 1' set to '60', 'procentuální mnoz. ceme' set to '25', and 'max. počet špatných bodu 2' set to '12'; and 'Ostatní' (Other) with checkboxes for 'Zobrazovat Histogram', 'Zobrazovat negativ', and 'Zobrazovat roztažení kon.'. The 'Nálezy' tab on the right shows a list of detected objects with their coordinates and percentages.

Object	X	Y	Percentage
25 bod	60 %		
mez:0			
x: min25	max:45		
y: min0	max:91		
1	20,634%	13	
1	79,365%	50	

Obr. 6.9 Hlavní záložky pro nastavení

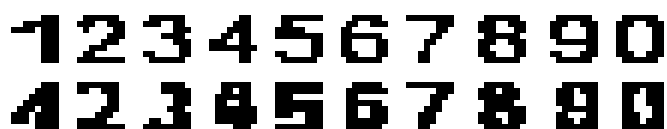
Záložka Graf slouží pro vykreslování původního digitalizovaného signálu přecházejícího z videokamery. Tento signál je zachycen pomocí skládání signálu a reprezentuje průběh na videokameře. V grafu je možno si přibližovat a oddalovat určitou část průběhu pomocí jednoduché práce s kurzorem. Na Obr. 6.10 je možno vidět vykreslený signál přichodící z videokamery. Jedná se o přiblížení dvou řádků signálu.



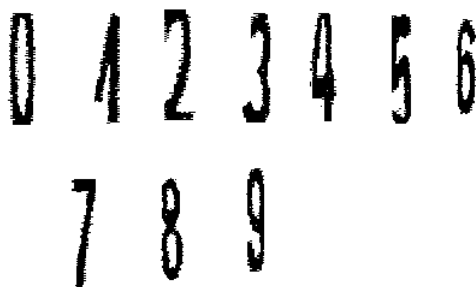
Obr. 6.10 Zobrazený signál z videokamery

6.5 Měření a vyhodnocení výsledku analýzy obrazu

Při testování byl použit model jednotlivých čísel vystříhnut z papíru a přilepený na okno, díky čemuž bylo zajištěna jistá úroveň kontrastu. Poté byly jednotlivé snímky nasnímány pomocí vytvořené aplikace pro snímání obrazu. Na Obr. 6.12 je možno vidět jednotlivé prvky nasnímané pomocí kamery po provedení prahování. Po využití programu byl vytvořen normalizovaný tvar jednotlivých číslic (Obr. 6.11) a následně byli získány hodnoty s pravděpodobnostní odchylkou od předlohy.



Obr. 6.11 Normalizovaný tvar získaný při měření č. 1 (dole) a předloha(nahoře)



Obr. 6.12 Příklad prahovaného obrazu nasnímaného kamerou

Číslo měření	1		2		3		4		5	
Číslo	E(p)	výsledek analýzi	E(p)	výsledek analýzi	E(p)	výsledek analýzi	E(p)	výsledek analýzi	E(p)	výsledek analýzi
-	%	-	%	-	%	-	%	-	%	-
0	27,0	0	39,7	8	39,7	8	38,1	0	20,6	0
1	39,7	4	28,6	1	39,7	4	25,4	1	34,9	1
2	20,6	2	25,4	2	33,3	2	27,0	2	31,8	2
3	38,1	8	33,3	3	32,0	3	25,4	3	33,3	3
4	33,3	4	34,9	4	34,9	4	44,4	4	25,4	4
5	38,1	5	34,9	5	38,1	5	38,1	5	44,4	6
6	23,8	6	25,4	6	33,3	6	36,5	8	33,3	6
7	7,9	7	22,2	7	32,8	7	25,4	7	28,6	7
8	23,8	8	31,8	8	25,4	8	27,0	8	44,4	6
9	27,0	9	25,4	9	33,3	9	34,9	0	34,9	8

Číslo měření	6		7		8		9		10	
Číslo	E(p)	výsledek analýzi	E(p)	výsledek analýzi	E(p)	výsledek analýzi	E(p)	výsledek analýzi	E(p)	výsledek analýzi
-	%	-	%	-	%	-	%	-	%	-
0	42,9	8	36,5	0	44,4	8	25,4	0	23,8	0
1	39,7	4	19,1	1	34,9	1	25,4	1	20,6	1
2	36,5	7	33,3	2	30,2	2	36,5	7	44,4	7
3	30,2	3	36,5	3	25,4	3	28,6	3	23,8	3
4	34,9	4	25,4	4	23,8	1	25,4	4	27,0	1
5	39,7	6	25,4	5	28,6	5	20,6	5	44,4	6
6	30,2	6	20,6	6	42,9	8	36,5	6	44,4	8
7	31,8	7	27,0	7	27,0	7	33,3	7	44,4	7
8	39,7	9	32,0	8	19,1	8	25,4	8	42,9	9
9	25,4	9	19,1	9	25,4	9	27,0	9	23,8	9

Tabulka 1 Naměřené hodnoty bílé čísla

Byly provedeny dvě kontroly. V Tabulka 1 jsou hodnoty získané při měření bílé předlohy nalepené na sklo. Při tomto měření docházelo k částečnému průniku světla a tím tak k malému rozdílu kontrastu. To mělo za následek nesprávné automatické zvolení prahu a tím také k vyšší chybivosti.

Při druhém měření byla zvolena černá předloha opět přilepená na okno. U toho to měření došlo k lepšímu kontrastu a tím také k lepší identifikaci daného čísla.

Číslo měření	1		2		3		4		5	
Číslo	E(p)	výsledek analýzi	E(p)	výsledek analýzi	E(p)	výsledek analýzi	E(p)	výsledek analýzi	E(p)	výsledek analýzi
-	%	-	%	-	%	-	%	-	%	-
0	22,2	0	27,0	0	28,6	0	34,9	0	23,8	0
1	15,9	1	17,5	1	20,6	1	15,9	1	39,7	4
2	19,1	2	15,9	2	44,4	7	19,1	2	20,6	2
3	28,6	3	34,9	3	32,0	3	23,8	3	38,1	8
4	32,0	4	34,9	4	44,4	1	38,1	1	33,3	4
5	30,2	5	15,9	5	38,1	6	33,3	5	32,8	5
6	25,4	6	19,1	6	36,5	6	32,8	6	38,1	8
7	22,2	7	23,8	7	25,4	7	33,3	7	33,3	7
8	15,9	8	25,4	8	22,2	8	32,8	8	32,8	8
9	25,4	9	31,8	9	25,4	9	25,4	9	33,3	9

Číslo měření	6		7		8		9		10	
Číslo	E(p)	výsledek analýzi	E(p)	výsledek analýzi	E(p)	výsledek analýzi	E(p)	výsledek analýzi	E(p)	výsledek analýzi
-	%	-	%	-	%	-	%	-	%	-
0	33,3	0	27,0	0	30,2	0	22,2	0	44,4	8
1	19,1	1	20,6	1	25,4	1	39,7	4	9,5	1
2	19,1	2	25,4	2	30,2	2	36,5	7	11,1	2
3	33,3	3	38,1	8	25,4	3	23,8	3	28,6	3
4	32,0	4	34,9	4	34,9	4	30,2	4	25,4	4
5	34,9	5	38,1	6	15,9	5	25,4	5	30,2	5
6	36,5	6	33,3	6	15,9	6	30,2	6	25,4	6
7	33,3	7	32,8	7	33,3	7	25,4	7	22,2	7
8	32,8	8	22,2	8	32,8	8	28,6	8	15,9	8
9	25,4	9	15,9	9	28,6	9	25,4	9	19,1	9

Tabulka 2 Naměřené hodnoty černá čísla

7 Závěr

Tato bakalářská práce byla zaměřena na monitorování a zpracování obrazu videosignálu s analogové kamery pomocí vestavěného systému tvořenou univerzální vývojové desky EvbDSP56800.

Algoritmy vytvořené ve vestavěném systému umožňují s ohledem na nedostatečnou paměť a rychlost vnitřního A/D převodníku, snímání obrazu v rozlišení 10x70 a postupného přeposílání získaných dat do počítače pomocí přídavného modulu USB, který byl navržen pro tuto bakalářskou práci. Také bylo možno pomocí vývojové desky EvbDSP56800 provést jednoduché automatické nalezení prahu a následně taktéž přeposílání těchto dat do počítače.

Díky přeposílání dat do počítače a využití postupného posouvání vzorkování v obraze bylo možno dosáhnout v počítači složení obrazu maximálního rozlišení 40x70. Toto vyplynulo z malé paměti a rychlosti vnitřního A/D převodníku. Pomocí jednoduchého algoritmu bylo možno toto rozlišení zvýšit při vykreslování v počítači přibližně na 80x70, což je již dostatečné pro provádění analýzy obrazu.

Analýza byla prováděna v počítači. V rámci bakalářské práce byl vytvořen program v programovacím jazyce C#, umožňujícího připojení pomocí USB modulu k vývojové desce EvbDSP56800 a získávání dat zachycovaných pomocí analogové kamery. Tyto data byla následně vykreslovány, upravovány a byla umožněna jejich analýza.

Signál získaný z analogové kamery mohl být převeden na obraz díky znalostem o analogovém videosignálu popsanych v kapitole 3. Vestavný systém pro monitorování je blíže popsán v kapitole 4, kde je vysvětlen postup, jak dochází k snímání obrazu z analogové videokamery následnému automatickému hledání prahu a odesílání získaných dat do počítače k analýze obrazu. Jasové transformace pro zpracovávání obrazu jsou vysvětleny v kapitole 5. Díky použití roztažení kontrastu vysvětleného v této kapitole bylo umožněno lepší analýzy obrazového signálu z kamery. Samotná analýza a její metody jsou popsány v kapitole 6, kde je také vysvětleno uživatelské rozhraní programu navrženého pro tuto bakalářskou práci a také provedeno měření za účelem ohodnocení využitých metod získávání a analýzy obrazového signálu z analogových kamer.

Díky snaze maximálně využít vývojovou desku EvbDSP56800 bylo nutno snížit nároky na snímání obraz, ten ale byl dostatečný pro následnou analýzu. Ovšem v případě lepšího rozlišení obrazu by docházelo ke zkvalitnění analýzy obrazu. Při měření a vyhodnocování analýzy obrazu bylo zjištěno, že je možno použitím metody automatického hledání prahu snímat a analyzovat obraz, který nemá přesně definovanou barvu. Tedy přesněji je možno analyzovat i objekty zájmu, které nemají zcela černou barvu. Jedinou podmínkou, která musí být dodržena je, aby objekt zájmu byl tmavší než pozadí. V takovém případě ovšem dochází k menšímu rozložení kontrastu mezi objektem zájmu a pozadí a dochází k větší chybovosti při analýze objektů.

Pro možnosti vylepšení stávajícího systému je možno navrhnout a připojit externí paměť. Ta by umožnila zvýšení řádkového rozšíření obrazu získávaného pomocí desky EvbDSP56800. Přidáním externího paralelního A/D převodníku s vysokou rychlostí by bylo možno zvýšit sloupcové rozlišení a tím také zvýšit rozlišení získávaného obrazu. Díky rozšíření paměti by poté bylo možno také implementovat algoritmy pro zpracování a analýzu obrazu, které jsou vesměs prováděny na počítači do vestavného systému tvořeného vývojovou deskou EvbDSP56800. Také by bylo možno použít digitální kameru, tedy by nebylo nutno využívat paralelního A/D převodníku.

8 Seznam použité literatury

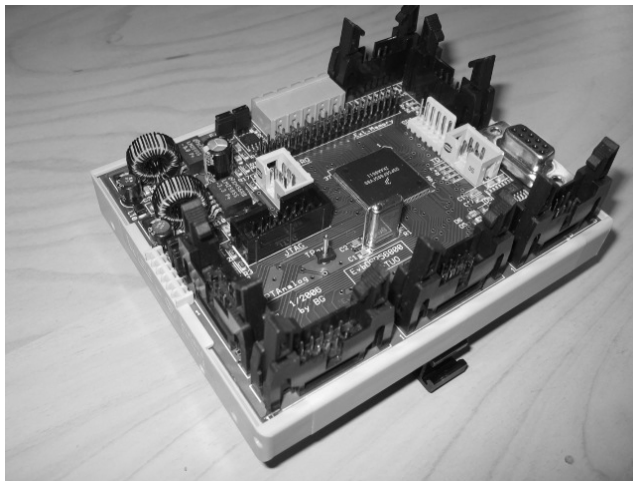
- [1] VÍT, V., KUBA, P. *Televizní technika : studiové zpracování televizního signálu*. 1.vyd. Praha : BEN technická literatura, 2000. 208 s. ISBN 80-86056-88-0.
- [2] DOBEŠ, M. *Zpracování obrazu a algoritmy v C#*. 1.vyd. Praha : BEN, 2008. 144 s. ISBN 978-80-7300-233-6.
- [3] BURKHARD, K. *USB : měření, řízení a regulace pomocí sběrnice USB*. 1. vyd. Praha : BEN technická literatura, 2002. 256 s., CD. PC & elektronika. ISBN 80-7300-073-3.
- [4] VÍT, V. *Televizní technika : přenosové barevné soustavy*. 1.vyd. Praha : BEN technická literatura, 1997. 720 s. ISBN 80-86056-04-X.
- [5] VÍT, V., GREGORA, P. *Televizní technika : Zařízení pro přenos a vysílání televizního signálu*. 1.vyd. Praha : BEN technická literatura, 2000. 176 s. ISBN 80-86056-89-9.
- [6] BEZDĚK, M. *Elektronika III*. 1. vyd. [s.l.] : Kopp, 2004. 236 s. ISBN 80-7232-241-9.
- [7] MACHÁČEK, Z., NEVŘIVA, P. *Číslicové signály a soustavy*. VŠB TU Ostrava, 2006.
- [8] HLAVÁČ, V., SEDLÁČEK, M., *Zpracování signálů a obrazů*. Praha : ČVUT, 2002. 220 s. ISBN 80-01-02114-9.
- [9] UHLÍŘ, J., SOVKA, P., *Číslicové zpracování signálů*. Praha : Vydavatelství ČVUT, 1995. 313 s.
- [10] <http://www.ftdichip.com>.
- [11] <http://www.freescale.com>.
- [12] Jiráček, E. Analýza obrazu - moderní prostředek pro technickou praxi. [Online] **2004**. http://www.odbornecasopisy.cz/index.php?id_document=32337 (accessed April 01, 2010).

9 Seznam příloh

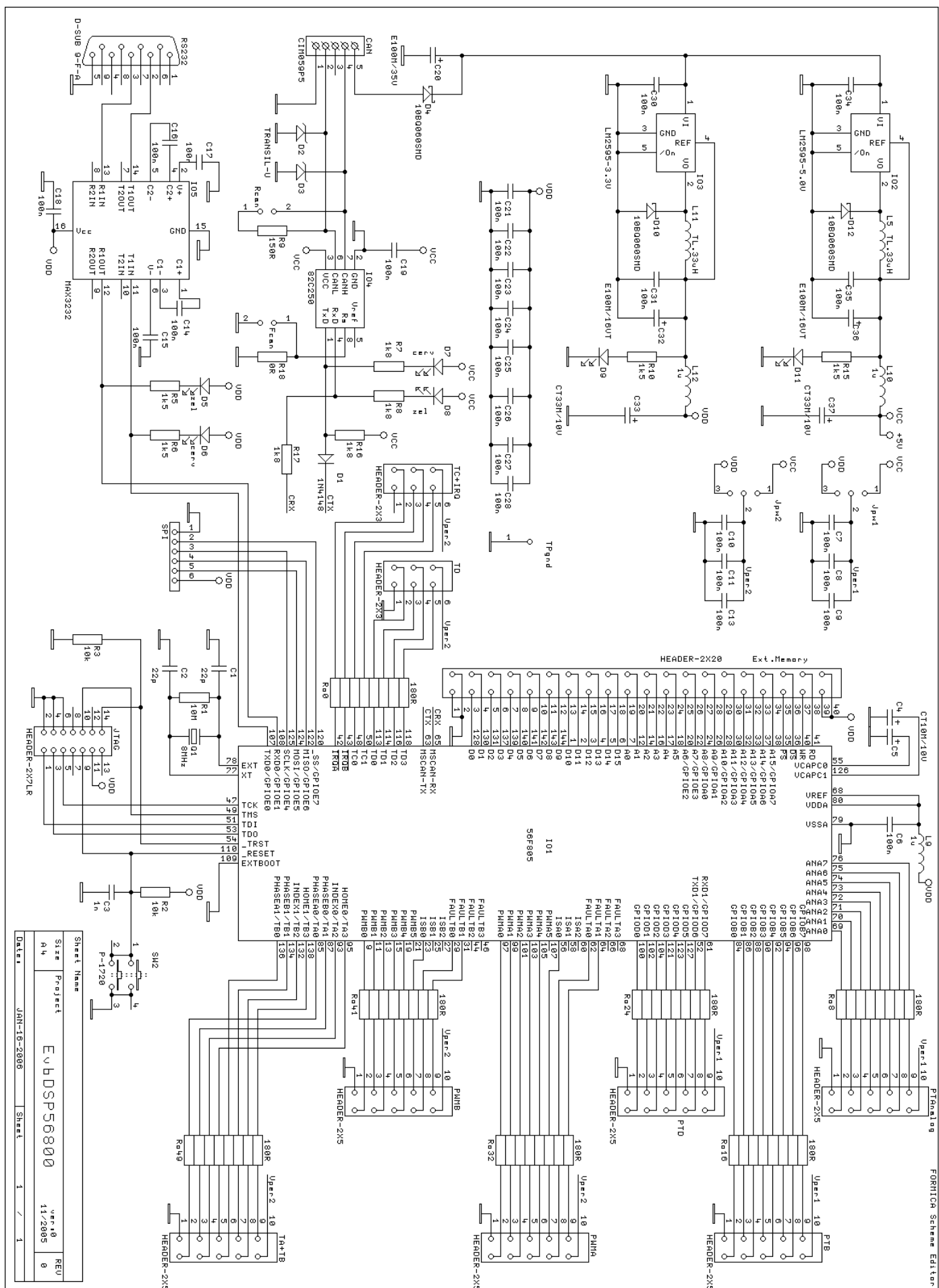
Příloha 1. - Univerzální vývojová deska EvbDSP56800.....	I
Příloha 2. - USB modul	III

Příloha 1. - Univerzální vývojová deska EvbDSP56800

Univerzální vývojová deska EvbDSP56800 je určena pro laboratorní účely. Schéma této univerzální desky je možno vidět na Obrázek 2. Tato deska k dispozici v laboratoři a jejím autorem je Ing. Kotzian Jiří, Ph.D.



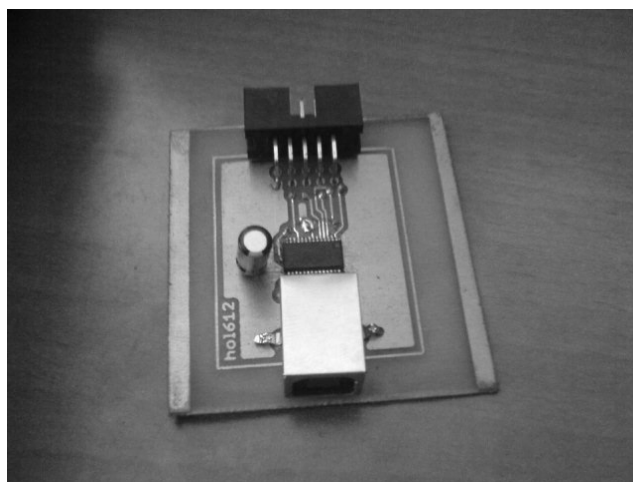
Obrázek 1 Deska EvbDSP56800



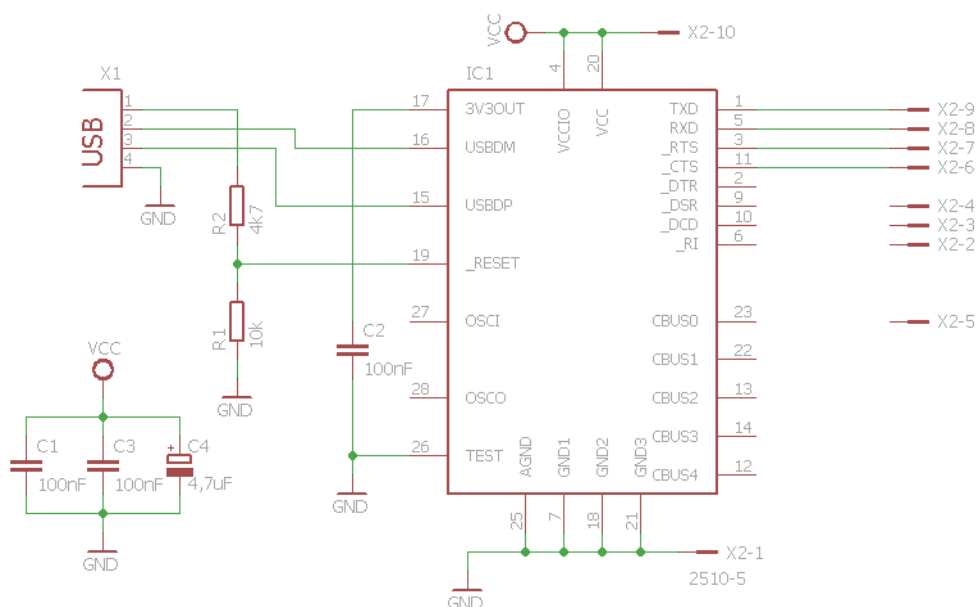
Obrázek 2 Schéma desky EVDSP56800

Příloha 2. - USB modul

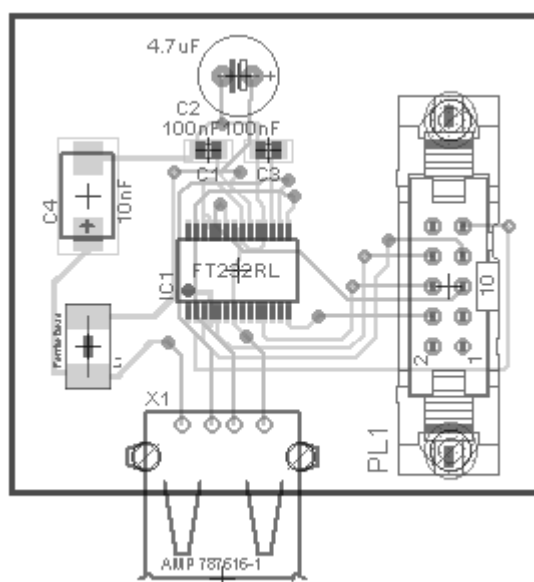
K vývojové desce EvbDSP56800 je připojen USB modul, který byl navrhnut pro tuto aplikaci a lze jej vidět Obrázek 3. Jeho schéma je možno vidět na Obrázek 4.



Obrázek 3 Deska USB modulu FT232RL



Obrázek 4 schéma vyvinutého USB modulu FT232RL



Obrázek 5 Deska vyvinutého USB modulu FT232RL